

**Nuclear magnetic resonance line shape of
 ^{29}Si in single-crystal silicon**

by

Brooks Christensen

Thesis defense date

March 10th, 2017

Thesis advisor: Prof. John C. Price

Department of Physics

Honors Council representative: Prof. Paul Beale

Department of Physics

Third reader: Prof. Mark Ablowitz

Department of Applied Mathematics

Christensen, Brooks

Nuclear magnetic resonance line shape of

^{29}Si in single-crystal silicon

Thesis directed by Prof. John C. Price

Measurements of the NMR line shape of 4.6832% abundant ^{29}Si in single-crystal silicon for four different crystallographic orientations relative to the applied magnetic field are reported. Significant inhomogeneous line broadening is avoided by immersing the sample crystals in a susceptibility matched liquid, and the proton NMR line shape in the liquid is used to measure the residual susceptibility mismatch and to shim the applied field. The observed line shapes are shown to be in good agreement with disorder-averaged spin dynamics simulations performed on a $4 \times 4 \times 4$ unit cell lattice with periodic boundary conditions. The splitting of resolved doublet features is used to measure the asymmetry ΔJ of the nearest-neighbor indirect dipole or J-coupling tensor, with the result $\Delta J = 90 \pm 20$ Hz. All resolved doublets can be identified with specific nearest and next-nearest neighbor isolated spin pairs that occur in the disorder ensemble.

Dedication

To Paris Souval, for giving me the chance to succeed. You are missed, my friend.

Acknowledgements

The author thanks Arthur Pardi and Richard Shoemaker for assistance with the spectrometer and Rahul Nandkishore, Ana Maria Rey, Charles Rogers, Ilya Kuprov and Jonathan Yates for helpful comments. This work was supported by an IGP seed grant from the University of Colorado, and by the Interdisciplinary Center for Electronics at the University of Colorado.

The author would also like to thank Professor John C. Price for teaching the necessary research skills to complete this thesis project and for his guidance and mentorship throughout the course of the research. The author would also like to thank Andrew Harriott, Anean Montgomery, Shawn Christensen, Lisa Christensen, Megan Beaudoin, Chris McRee, Nathan Turner, Lindsay Christensen, Julie Bourgeois, Travis Bengel, Juliet Pilewskie, Jarred Heath, and Adrian Coronado for their history of support.

Contents

Chapter

1	Introduction	1
1.1	Solid state interactions in NMR	2
1.2	Thesis road map	4
2	The quantum mechanics of solid state NMR spectroscopy	6
2.1	The density operator	6
2.2	The NMR Hamiltonian	8
2.2.1	The Zeeman term	8
2.2.2	The dipole-dipole term	9
2.2.3	The J-coupling term	12
2.2.4	Transformation to the rotating frame	14
2.3	The secular approximation	15
2.3.1	Dipole-dipole interaction approximation	15
2.3.2	J-coupling interaction approximation	18
2.4	Quantum spin dynamics	24
2.4.1	The initial state of the system	24
2.4.2	Relaxation and the time constants T_1 and T_2	25
3	Materials	26
3.1	Why silicon?	26

3.2	Properties of Silicon	27
3.3	Susceptibility matching	30
3.3.1	Modeling and testing the susceptibility mismatch	34
3.4	Sample construction	37
3.5	X-ray diffractometry	39
4	Simulations using Spinach	40
4.1	Rotations and orientations	40
4.2	Lattice construction	43
4.2.1	Disorder Averaging	47
4.3	Simulation properties and organization	49
4.3.1	Including the J-coupling interaction	53
4.3.2	Processing the FIDs and plotting	54
5	Experiment and Analysis	55
5.1	The pulse-acquire experiment	55
5.2	T_1 measurement	56
5.3	Line shape of ^{29}Si in single-crystal silicon	57
5.3.1	Derivation of splitting size	59
5.3.2	Effect of the J-coupling interaction	59
5.4	Comparison of experimental results to model	64
5.4.1	Model Convergence	64
6	Conclusion and results	68

Bibliography **70**

Appendix

A	UNITY 500 MHz spectrometer operation	72
A.1	Setting up an experiment	72
B	Bruker X-ray diffractometer operation	76
B.1	Sample Preparation	76
B.2	Starting the Bruker	76
B.3	Acquiring rotation data with SMART	77
B.4	To turn off	78
B.5	Using fit2d to create a tiff image	78
C	The crystal rotation method	81
D	Radia code	83
E	Spinach code	93
E.1	Silicon simulation main body code	93
E.2	Lattice builder function code	100
E.3	J-coupling interaction function code	105
E.4	FID processing and spectrum plotting code	109

Tables

Table

5.1	Bond lengths, angles, and splitting frequencies in the 100 and 111 orientations . . .	61
5.2	Bond lengths, angles, and splittings in the 110 and 112 orientations	61
B.1	Bruker voltage increment durations	77

Figures

Figure

1.1	Diagram of spectrometer	3
3.1	The silicon conventional unit cell	28
3.2	The silicon primitive unit cell	29
3.3	The conventional Silicon unit cell and four experimental orientations	31
3.4	Radia model of the experimental sample geometry	35
3.5	Radia model of proton line shape of susceptibility mismatch	36
3.6	Diagram of constructed sample	38
4.1	Three-dimensional plot of simulated cluster	48
4.2	FIDs generated by single configurations	50
5.1	T1 Measurements	58
5.2	Experimentally obtained line shapes in all four orientations with simulated line shapes for comparison	60
5.3	Frequency splitting stem plots for all four orientations	62
5.4	J-coupling effect	63
5.5	Disorder average convergence	66
5.6	Basis level convergence	66
5.7	Cell size convergence	67

Chapter 1

Introduction

The subject of this thesis is the line shape produced by nuclear magnetic resonance (NMR) spectroscopy when performed on a single-crystal silicon sample in a magnetic susceptibility matched environment. This thesis is an extension of a paper recently submitted for publication by the current author and Professor Price [1]. The long-term goal of the project is to look for evidence of many-body localization [2] in disordered nuclear spin systems which might be revealed through spin-transport experiments. In this preliminary experiment the NMR line shape is explored to refine our knowledge of the spin Hamiltonian and to test our simulation methods. Because the NMR-active spin-1/2 isotope of silicon is dilute in the crystalline sample, this is a very well-defined disordered interacting spin system.

NMR spectroscopy has been used as a technique for materials analysis since Felix Bloch at Stanford University and Edward Mills Purcell at Harvard University independently developed the technique and were awarded the Nobel prize in physics in 1952. There are many different types of NMR experiments, mostly involving liquid samples [3, 4]; however, the focus of this experiment is solid-state NMR, where magnetic dipole interactions play an important role. A magnetic susceptibility matching method to reduce line broadening will be presented herein which has not been reported in previous experiments involving solid-state NMR spectroscopy of single-crystal silicon [5, 6, 7, 8, 9, 10].

The spectrometer is used to observe the time evolution of the nuclear magnetization of the sample. In this project the spectrometer is set up as depicted in Fig. 1.1. The superconducting

solenoid producing the static external magnetic field is surrounded by liquid helium and then a layer of liquid nitrogen all housed inside a vacuum sealed dewar. Before the experiment, the sample tube is loaded into a room temperature cavity at the center of the dewar. The sample tube is surrounded immediately by the RF coils which are wrapped in a saddle configuration about the tube to produce a transverse field despite its vertical central opening. The shim coils surround the RF coils and provide fine adjustments to the static magnetic field. (See Sec. 3.3 for more about the role of shim coils in this project.) The computer controls the RF coils which can switch between transmit and receive modes. The pulse-acquire experiment in this project transmits an RF pulse to manipulate the spins in the material and then receives the resulting magnetization signal produced in the sample. The excited transverse magnetization is the observable in this NMR experiment.

1.1 Solid state interactions in NMR

In general NMR experiments, there are different types of interactions which can be involved which are broadly grouped into the two general categories: magnetic and electric interactions. The magnetic interactions include spin-spin couplings and the chemical shift. The spin-spin couplings are interactions between magnetic moments of the spin-1/2 nuclei; these couplings are the primary interactions of interest in this project. The direct magnetic spin-spin coupling will be referred to as the dipole-dipole interaction, and the indirect magnetic spin-spin coupling will be referred to as J-coupling in this report. The chemical shift can be separated into an isotropic part and an anisotropic part which will both be discussed in Sec. 2.2. The only electric interaction is the quadrupole coupling, which is not applicable here since it involves nuclei with spin greater than 1/2; the material used in this project is deliberately chosen to include only spin-0 (non-magnetic) and spin-1/2 nuclei.

In isotropic liquids, free molecular rotations in three-dimensions average away dipole-dipole interactions and the anisotropic part of the chemical shift in NMR experiments; the only important interactions in liquids are the chemical shift and the indirect J-coupling. The primary differences between typical NMR experiments and those of the type performed in this project are the strengths

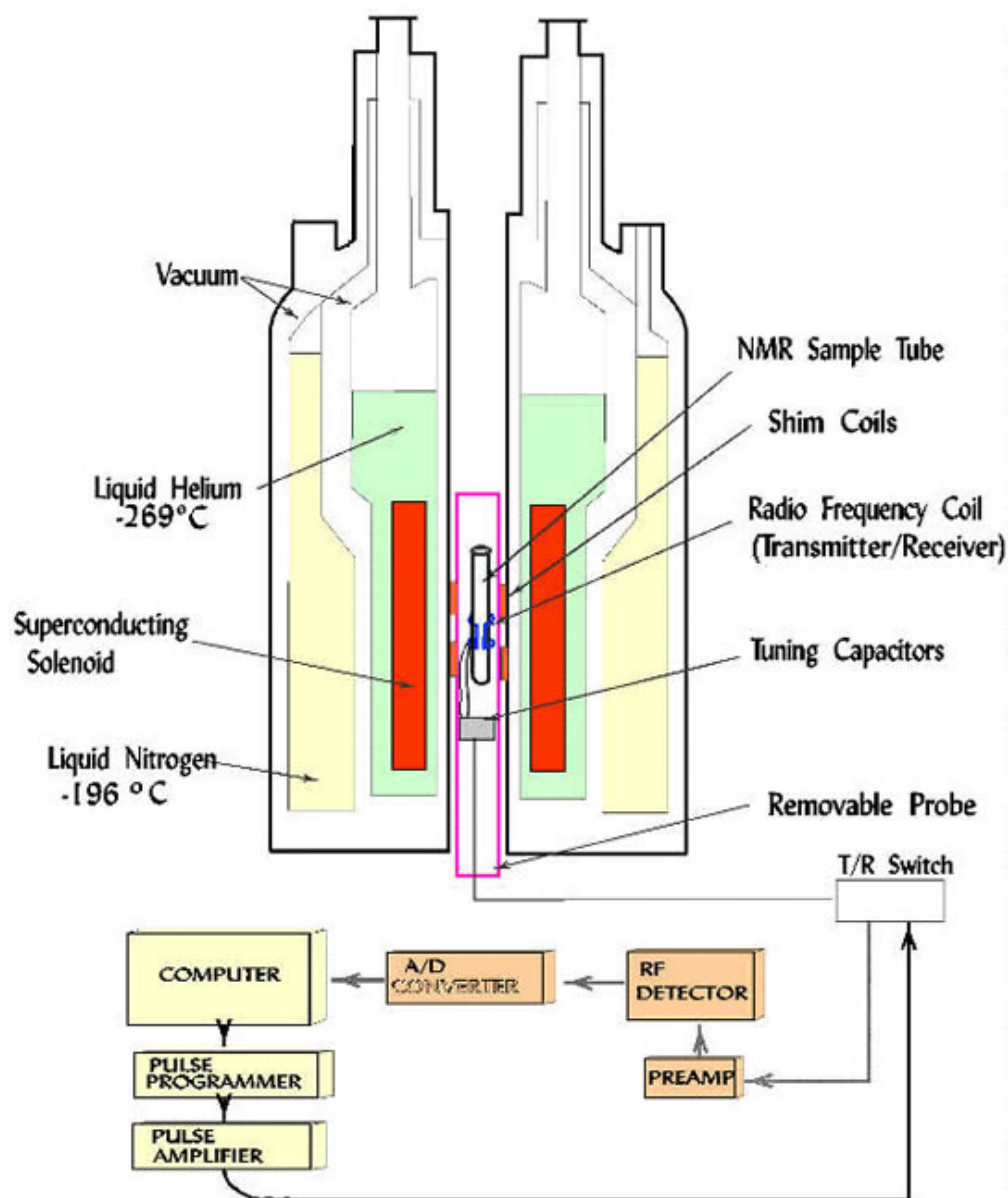


Figure 1.1: A diagram of the NMR spectrometer similar in design to the one used in this project. This figure is reproduced from <http://lucas.lakeheadu.ca/luil/nuclear-magnetic-resonance-nmr-facility/>.

of the pertinent atomic interactions. These interactions will be discussed in detail in Ch. 2.

Single-crystal silicon is a member of the diamond crystal class, which has no chemical shift anisotropy; only the isotropic part of the chemical shift will be described in Ch. 2 when these atomic interactions are discussed. Single-crystal silicon has no stable isotopes with spin greater than $1/2$. Coherent quantum systems have been proposed using silicon and diamond [11, 12, 13, 14, 15] as the material of interest.

The only active atoms in the silicon lattice are those of the ^{29}Si isotope. The natural abundance for this isotope is 4.6832%, which is considered to be in the dilute regime. The crystal lattice is ordered in space with periodic translational symmetry, but each atomic site in the lattice is randomly occupied by any one of the three stable isotopes of silicon (^{28}Si , ^{29}Si , and ^{30}Si). Therefore, the disorder in the lattice can be characterized by the single isotopic abundance parameter.

1.2 Thesis road map

This thesis is structured according to the following outline:

- Chapter 2 the Hamiltonian and quantum mechanics for the specific system used in this project will be discussed.
- The properties of single-crystal silicon will then be described in Ch. 3. The details of the sample construction and a description of a technique to improve on field uniformity in the experiment are also given in Ch. 3.
- Simulations used to model the primary features of the line shape from our solid-state NMR spectroscopy experiment are described in Ch. 4. Rotations are discussed in a general context and the statistics of lattice occupation are also discussed in the same chapter.
- The experiment is discussed in detail including results obtained in Ch. 5. Data analysis for the line shape, a measurement of the J-coupling parameters, and other structural features of the line shape are also discussed in the same chapter.

- The conclusion and a discussion of experimental results is given in Ch. 6, and some applications and extensions of this work are also presented.

Chapter 2

The quantum mechanics of solid state NMR spectroscopy

In this chapter the quantum mechanical foundation of NMR spectroscopy in solids will be introduced. Standard texts for NMR spectroscopy include Refs. [3, 4]. Introductory quantum mechanics and statistical mechanics texts include Refs. [16, 17, 18, 19].

2.1 The density operator

The system of interest in this project is a macroscopic collection of identical interacting spin-1/2 nuclei. For an ensemble of many interacting particles, the density operator $\hat{\rho}$ describes the complete state of the system and may be used to calculate both thermal and quantum averages. The energy eigenstate with energy E_i of the system state may be represented in Dirac notation as $|i\rangle$, and the eigenvalue equation gives

$$\hat{H}|i\rangle = E_i|i\rangle. \quad (2.1)$$

The density operator for an ensemble of energy eigenstates is

$$\hat{\rho} \equiv \sum_i \mathcal{P}_i |i\rangle\langle i| \quad (2.2)$$

where \mathcal{P}_i is the probability for the system to be found in the state $|i\rangle$ corresponding to system energy E_i .

According to Boltzmann statistics, the probability for a system at thermal equilibrium to be found with energy E_i is given by

$$\mathcal{P}_i = \frac{e^{-E_i/kT}}{Z}, \quad (2.3)$$

where Z is the system partition function given by

$$Z = \sum_i e^{E_i/kT}. \quad (2.4)$$

The partition function is the sum of the Boltzmann factors associated with each energy eigenstate.

The Boltzmann factors can be related to the Hamiltonian \hat{H} by

$$e^{-\hat{H}/kT} = \sum_i e^{-E_i/kT} |i\rangle\langle i| \quad (2.5)$$

since each matrix element is identical on the left and right hand sides. The trace of this matrix is now the partition function, since

$$\text{Tr}(e^{-\hat{H}/kT}) = \sum_j \langle j|e^{-\hat{H}/kT}|j\rangle = \sum_i e^{-E_i/kT} = Z. \quad (2.6)$$

Therefore, using Eqs. 2.3 and 2.4 we can now express the thermal equilibrium density operator as

$$\hat{\rho} \equiv \frac{1}{Z} \sum_i e^{-E_i/kT} |i\rangle\langle i| = \frac{e^{-\hat{H}/kT}}{\text{Tr}(e^{-\hat{H}/kT})}. \quad (2.7)$$

We will be concerned with the high temperature limit where the exponent of Eq. 2.7 is a small quantity (details of the structure of the Hamiltonian are addressed in Sec. 2.2). We may Taylor expand to obtain

$$\hat{\rho} \approx \frac{1}{Z} (\mathbb{I} - \frac{\hat{H}}{kT}). \quad (2.8)$$

In most cases we can ignore the identity operator because it will not contribute to the expectation values of interest.

As mentioned above, the utility of the density operator is for its role in the calculation of thermally averaged expectation values. The expectation value of an arbitrary operator \hat{O} is

$$\langle \hat{O} \rangle = \text{Tr}(\hat{O}\hat{\rho}), \quad (2.9)$$

which can be seen by plugging in the definition for $\hat{\rho}$ from Eq. 2.2. This substitution yields

$$\langle \hat{O} \rangle = \text{Tr}(\hat{O} \sum_i \mathcal{P}_i |i\rangle\langle i|) = \sum_j \langle j|\hat{O} \sum_i \mathcal{P}_i |i\rangle\langle i|j\rangle. \quad (2.10)$$

The term $\langle j|i \rangle$ is the Kronecker delta with

$$\langle j|i \rangle = \delta_{i,j}, \quad \delta_{i,j} \equiv \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases} \quad (2.11)$$

This factor selects out terms from the sum over i in which $i=j$, so there is only one surviving term for each value of j . This collapses the double sum into a sum over j only, or

$$\langle \hat{O} \rangle = \sum_j \langle j|\hat{O}\mathcal{P}_j|j \rangle = \sum_j \mathcal{P}_j \langle j|\hat{O}|j \rangle, \quad (2.12)$$

since \mathcal{P}_j is a constant for each value of j .

2.2 The NMR Hamiltonian

The NMR Hamiltonian for a collection of identical spin-1/2 nuclei in a solid can be expressed as the sum of three terms corresponding to three different interactions [20]. These interactions in order of descending strength are: 1) the Zeeman interaction due to the externally applied magnetic field, 2) the dipole-dipole interaction between two nuclear magnetic dipoles, and 3) the indirect dipole or J-coupling interaction. The Hamiltonian can be expressed as

$$\hat{H} = \hat{H}_Z + \hat{H}_D + \hat{H}_J, \quad (2.13)$$

Each term will be explored in detail below.

2.2.1 The Zeeman term

The dominant term in the NMR Hamiltonian corresponds to the Zeeman interaction which for the i^{th} spin in the system is given by

$$\hat{H}_{Z,i} = -\vec{\mu}_i(\mathbb{I} + \overleftrightarrow{\sigma})\vec{B}_0, \quad (2.14)$$

where $\vec{\mu}$ is the magnetic dipole moment and $\overleftrightarrow{\sigma}$ is the chemical shift tensor specific to the material.

For silicon the chemical shift is isotropic and the same for all spins; the only effect of an isotropic

chemical shift is to slightly change the effective applied field. The interaction Hamiltonian then becomes

$$\hat{H}_{Z,i} = -\vec{\mu}_i \cdot \vec{B}_{0,\text{effective}}. \quad (2.15)$$

The quantum magnetic dipole moment $\vec{\mu}$ is given in terms of the gyromagnetic ratio γ and the spin operator \vec{S} as

$$\vec{\mu} = \gamma \vec{S}, \quad (2.16)$$

where \vec{S} is the spin vector operator given by $\vec{S} = S^x \hat{x} + S^y \hat{y} + S^z \hat{z}$. For ^{29}Si $\gamma_{Si} = -8.465 \cdot 2\pi$ MHz/T. We express terms in \hat{H} in angular frequency units $2\pi \cdot \text{Hz}$ and assume the externally applied magnetic field to be aligned with the z-axis

$$\vec{B}_0 = B_0 \hat{z}. \quad (2.17)$$

The dot product between $\vec{\mu}$ and \vec{B}_0 in Eq. 2.15 selects out only the z-component of the spin operator, and yields the Zeeman Hamiltonian term for a single spin as

$$\hat{H}_{Z,i} = -\gamma_{Si} B_{0,\text{effective}} \hat{S}_i^z. \quad (2.18)$$

To obtain the Hamiltonian for the full system, a sum is taken over all spins yielding

$$\boxed{\hat{H}_Z = - \sum_i \gamma_{Si} B_{0,\text{effective}} \hat{S}_i^z = -\omega_L \hat{M}^z,} \quad (2.19)$$

where $\hat{M}^z = \sum_i S_i^z$ is the magnetization induced by the effective applied field and $\omega_L = \gamma_{Si} B_0$ is the Larmor frequency.

2.2.2 The dipole-dipole term

The dipole-dipole interaction is between two nuclear magnetic dipoles and is a much weaker interaction than the Zeeman interaction. It is also the primary interaction of interest in this project because it causes the initially uncorrelated thermal state to evolve into a state with many-spin correlations. The energy of interaction between two magnetic dipoles can be derived by the methods of classical electrodynamics [21] as follows.

The magnetic dipole is modeled similarly to an electric dipole where there are two magnetic charges of charge $\pm q_m$ separated by a distance $\delta \vec{r}$. The magnetic dipole moment is then defined to be

$$\vec{\mu} = q_m \delta \vec{r}. \quad (2.20)$$

If the negative magnetic charge ('south' end of dipole) is placed at the origin, then the magnetic field at any point \vec{r} in space is given by

$$\begin{aligned} \vec{B}_{-q_m} &= \frac{\mu_0}{4\pi} \frac{-q_m}{r^3} \vec{r} \quad \text{and} \\ \vec{B}_{+q_m} &= \frac{\mu_0}{4\pi} \frac{q_m}{|\vec{r} - \delta \vec{r}|^3} (\vec{r} - \delta \vec{r}), \quad \text{so} \\ \vec{B}_{total} &= \frac{\mu_0 q_m}{4\pi} \left(\frac{-\vec{r}}{r^3} + \frac{\vec{r} - \delta \vec{r}}{|\vec{r} - \delta \vec{r}|^3} \right), \end{aligned} \quad (2.21)$$

where $r = |\vec{r}|$, and $|\vec{r} - \delta \vec{r}|$ is the separation distance between the two magnetic charges. In the limit where the separation distance between the point of interest and the origin is much greater than the separation distance between the two magnetic charges (or $r \gg |\vec{r} - \delta \vec{r}|$), then $|\vec{r} - \delta \vec{r}| \approx r - (\vec{r} \cdot \delta \vec{r})/r \equiv r - \epsilon$. We may Taylor expand about $\epsilon = 0$:

$$\begin{aligned} f(\epsilon) &= \frac{1}{(r - \epsilon)^3} \approx f(\epsilon = 0) + f'(\epsilon = 0)\epsilon \\ &= \frac{1}{r^3} + \frac{(-3)(-1)}{(r - \epsilon)^4} \bigg|_{\epsilon=0} \epsilon \\ &= \frac{1}{r^3} + \frac{3}{r^4} \epsilon \\ &= \frac{1}{r^3} \left(1 + \frac{3}{r} \epsilon \right) \\ &\Rightarrow \frac{1}{|\vec{r} - \delta \vec{r}|^3} \approx \frac{1}{r^3} \left(1 + 3 \frac{\vec{r} \cdot \delta \vec{r}}{r^2} \right) \end{aligned} \quad (2.22)$$

With this simplification in hand, the first order approximation for the magnetic field at a large distance from the magnetic dipole is given by

$$\begin{aligned} \vec{B} &= \frac{\mu_0 q_m}{4\pi} \left[-\frac{\vec{r}}{r^3} + \frac{1}{r^3} \left(1 + 3 \frac{\vec{r} \cdot \delta \vec{r}}{r^2} \right) (\vec{r} - \delta \vec{r}) \right] \\ &= \frac{\mu_0 q_m}{4\pi} \frac{1}{r^3} \left(3 \frac{\vec{r} \cdot \delta \vec{r}}{r^2} \vec{r} - \delta \vec{r} \right) \\ \vec{B} &= \frac{\mu_0}{4\pi} \frac{1}{r^3} \left(3 \frac{\vec{r} \cdot \vec{\mu}}{r^2} \vec{r} - \vec{\mu} \right). \end{aligned} \quad (2.23)$$

The energy of a magnetic moment $\vec{\mu}$ in an externally applied magnetic field \vec{B} can be modeled by using two equations: one for the torque $\vec{\tau}$ of a magnetic dipole in the presence of an external field and one for relating the torque to the energy. The torque is

$$\vec{\tau} = \vec{\mu} \times \vec{B}, \quad (2.24)$$

and the energy can be found by integrating the torque through an angle necessary to bring the magnetic dipole into alignment with the externally applied magnetic field. This can be expressed as

$$E = \int_{\theta_0}^{\theta_1} |\vec{\tau}| d\theta. \quad (2.25)$$

Assume for simplicity with no loss of generality $\theta_0 = \frac{\pi}{2}$. The energy is then found to be

$$\begin{aligned} E &= \int_{\frac{\pi}{2}}^{\theta_1} |\vec{\mu} \times \vec{B}| d\theta \\ &= \int_{\frac{\pi}{2}}^{\theta_1} |\vec{\mu}| |\vec{B}| \sin \theta d\theta \\ &= |\vec{\mu}| |\vec{B}| \int_{\frac{\pi}{2}}^{\theta_1} \sin \theta d\theta \\ &= |\vec{\mu}| |\vec{B}| (-\cos \theta) \Big|_{\theta=\frac{\pi}{2}}^{\theta_1} \\ &= -|\vec{\mu}| |\vec{B}| \left[\cos \theta_1 - \cos \left(\frac{\pi}{2} \right) \right] \\ &= -|\vec{\mu}| |\vec{B}| \cos \theta_1, \end{aligned} \quad (2.26)$$

with θ_1 being the angle of separation between the two vectors $\vec{\mu}$ and \vec{B} . Note that this satisfies the definition of a vector dot product; thus, the energy may be expressed as

$$E = -\vec{\mu} \cdot \vec{B}. \quad (2.27)$$

Substituting the expression for \vec{B} from Eq. 2.23, and assuming that the field is being generated by $\vec{\mu}_2$ which acts on $\vec{\mu}_1$ to produce

$$\begin{aligned} E &= -\vec{\mu}_1 \cdot \vec{B} \\ &= -\vec{\mu}_1 \cdot \left[\frac{\mu_0}{4\pi} \frac{1}{r^3} \left(3 \frac{\vec{r} \cdot \vec{\mu}_2}{r^2} \vec{r} - \vec{\mu}_2 \right) \right] \\ &= -\frac{\mu_0}{4\pi} \frac{1}{r^5} [3(\vec{\mu}_1 \cdot \vec{r})(\vec{r} \cdot \vec{\mu}_2) - r^2 \vec{\mu}_1 \cdot \vec{\mu}_2]. \end{aligned} \quad (2.28)$$

The quantum definition of the magnetic dipole moment is $\vec{\mu}_i \equiv \gamma_i \hbar \vec{S}_i$, and the units of angular frequency are desired for the energy expression. This is accomplished by dividing the normal expression for energy (measured in Joules in SI) by a factor of \hbar . Using these substitutions and dividing through by the aforementioned factor of \hbar yields the Hamiltonian operator for the quantum dipole-dipole interaction:

$$\hat{H}_{DD} = -\frac{\mu_0 \gamma_1 \gamma_2 \hbar}{4\pi r^5} [3(\vec{S}_1 \cdot \vec{r})(\vec{r} \cdot \vec{S}_2) - r^2 \vec{S}_1 \cdot \vec{S}_2]. \quad (2.29)$$

In this particular project, the interactions are all homonuclear dipole-dipole interactions, so that $\gamma_1 = \gamma_2 = \gamma$. The dipole-dipole interaction Hamiltonian between any two particles i and j may now be expressed as

$$\hat{H}_{DD} = -\frac{\mu_0 \gamma^2 \hbar}{4\pi r_{i,j}^5} [3(\vec{S}_i \cdot \vec{r})(\vec{r} \cdot \vec{S}_j) - \vec{S}_i \cdot \vec{S}_j] \quad (2.30)$$

where $r_{i,j}$ is the separation distance between the two interacting nuclei. The notation may be simplified by defining the constant

$$b_{i,j} = \frac{\mu_0 \gamma^2 \hbar}{4\pi r_{i,j}^3}, \quad (2.31)$$

with units of angular frequency so that when a sum is taken over all particles in the system the dipole-dipole interaction Hamiltonian may finally be expressed as

$$\boxed{\hat{H}_{DD} = \sum_{i < j} b_{i,j} \left(\vec{S}_i \cdot \vec{S}_j - \frac{3(\vec{S}_i \cdot \vec{r}_{i,j})(\vec{S}_j \cdot \vec{r}_{i,j})}{r_{i,j}^2} \right)}. \quad (2.32)$$

2.2.3 The J-coupling term

The indirect magnetic dipole interaction, or J-coupling, term represents the effect of the electronic shielding between pairs of spins in the solid. This effect is the weakest interaction of the three presented here, but it still has a significant effect on the evolution of the system. The Hamiltonian in angular frequency units is

$$\boxed{\hat{H}_J = \sum_{i < j} 2\pi \vec{S}_i \overset{\leftarrow}{J} \vec{S}_j} \quad (2.33)$$

where \overleftrightarrow{J} is the J-coupling tensor between spins i and j and can be represented in matrix notation as

$$\overleftrightarrow{J} = \begin{bmatrix} J'_{xx} & J'_{xy} & J'_{xz} \\ J'_{yx} & J'_{yy} & J'_{yz} \\ J'_{zx} & J'_{zy} & J'_{zz} \end{bmatrix}. \quad (2.34)$$

The primes after each tensor label given above indicate that the elements are not in the principal axis basis but, rather, in some other general coordinate basis.

The strength of the J-coupling interaction falls off quickly with bond order and is weaker than the other interactions. For these reasons only the nearest-neighbor J-coupling interaction is included in this derivation.

In the silicon structure the lattice has an inversion center at the midpoint of each nearest-neighbor bond, the bond is a 3-fold rotation axis, and 3 mirror planes intersect along the bond. Because of these symmetries, the nearest-neighbor \overleftrightarrow{J} is a symmetric matrix and it has axial symmetry with one principle axis along the bond and two equal axes perpendicular to the bond [1]. The J-coupling tensor may then be represented as a matrix of the form

$$\overleftrightarrow{J} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (2.35)$$

with $J_{xx} = J_{yy}$, when expressed in the principal axis frame with the Cartesian \hat{z} unit vector aligned along the nearest-neighbor bond. A typical convention used to describe an axial J-coupling tensor uses J for the isotropic contribution and ΔJ for the asymmetrical contribution. These quantities are defined in this context to be

$$\begin{aligned} J &= \frac{1}{3}(J_{xx} + J_{yy} + J_{zz}) \\ \Delta J &= J_{zz} - \frac{1}{2}(J_{xx} + J_{yy}) \end{aligned} \quad (2.36)$$

2.2.4 Transformation to the rotating frame

The Zeeman Hamiltonian $H_0 = -\omega_L M^z$ causes all spins to precess at the Larmor frequency. Transforming to an interaction representation with respect to H_0 corresponds to transforming to a frame of reference rotating at the Larmor frequency.

The time dependent Schroedinger equation for the system specified by the Hamiltonian $\hat{H} = H_0 + H'$ is

$$i \frac{\partial}{\partial t} |\psi\rangle = (H_0 + H') |\psi\rangle. \quad (2.37)$$

Multiplying both sides by $e^{+iH_0 t}$ and multiplying by a factor of 1 on the right hand side yields

$$e^{+iH_0 t} \left(i \frac{\partial}{\partial t} \right) |\psi\rangle = e^{+iH_0 t} (H_0 + H') e^{-iH_0 t} e^{+iH_0 t} |\psi\rangle. \quad (2.38)$$

The left hand side of this equation can be written as $i \frac{\partial}{\partial t} (e^{+iH_0 t} |\psi\rangle) + H_0 e^{+iH_0 t} |\psi\rangle$ which produces

$$\begin{aligned} i \frac{\partial}{\partial t} (e^{+iH_0 t} |\psi\rangle) + H_0 e^{+iH_0 t} |\psi\rangle &= e^{+iH_0 t} (H_0 + H') e^{-iH_0 t} e^{+iH_0 t} |\psi\rangle \\ i \frac{\partial}{\partial t} (e^{+iH_0 t} |\psi\rangle) &= e^{+iH_0 t} H' e^{-iH_0 t} e^{+iH_0 t} |\psi\rangle. \end{aligned} \quad (2.39)$$

If we define $|\tilde{\psi}\rangle \equiv e^{+iH_0 t} |\psi\rangle$ and $\tilde{H}' \equiv e^{+iH_0 t} H' e^{-iH_0 t}$, then Eq. 2.39 can be written as

$$i \frac{\partial}{\partial t} |\tilde{\psi}\rangle = \tilde{H}' |\tilde{\psi}\rangle. \quad (2.40)$$

In the interaction representation, H_0 disappears from \hat{H} and all operators rotate about the z -axis. If O is an operator in the lab frame, then the transformed operator is

$$\tilde{O} = e^{-i\omega_L M^z} O e^{+i\omega_L M^z}. \quad (2.41)$$

Please note that the operator ‘hats’ will be dropped moving forward to reduce notational clutter. New operators will be specified explicitly if it is not clear from the context.

The density operator is defined to be $\rho = |\psi\rangle\langle\psi|$ and can be represented in the rotating frame as $\tilde{\rho} \equiv e^{+iH_0 t} \rho e^{-iH_0 t} = |\tilde{\psi}\rangle\langle\tilde{\psi}|$. The expectation value of O can be found by

$$\begin{aligned} \langle O \rangle &= Tr(O\rho) \\ &= Tr(e^{+iH_0 t} O e^{-iH_0 t} e^{+iH_0 t} \rho e^{-iH_0 t}) \\ &= Tr(\tilde{O} \tilde{\rho}). \end{aligned} \quad (2.42)$$

2.3 The secular approximation

The externally applied magnetic field is significantly stronger than any of the other fields considered in this experiment, so a simplification can be made to the Hamiltonian for the system. This simplification, known as the ‘secular approximation’ in the literature [20], can be made since the ^{29}Si Larmor frequency (≈ 100 MHz) is much greater than the (≈ 1 kHz) nearest-neighbor dipole interaction. In the rotating frame, terms which oscillate at the Larmor frequency or twice the Larmor frequency are dropped from the Hamiltonian since the evolution due to such terms is small compared to evolution due to those terms which do not oscillate.

2.3.1 Dipole-dipole interaction approximation

The dipole-dipole interaction can also be approximated in the rotating frame about the \hat{z} -axis. The components of the spherical coordinate system can be transformed into the Cartesian coordinate basis by the following set of equations (See figure):

$$\begin{aligned} r_z &= r \cos \theta \\ r_x &= r \sin \theta \cos \phi \\ r_y &= r \sin \theta \sin \phi. \end{aligned} \tag{2.43}$$

Now the dipole-dipole interaction Hamiltonian from Eq. 2.32 can be expressed in the Cartesian coordinate basis as

$$\begin{aligned} H_{DD} = -b_{i,j} [3(S_{1,x} \sin \theta \cos \phi + S_{1,y} \sin \theta \sin \phi + S_{1,z} \cos \theta)(S_{2,x} \sin \theta \cos \phi + S_{2,y} \sin \theta \sin \phi + S_{2,z} \cos \theta) \\ - (S_{1,x} S_{2,x} + S_{1,y} S_{2,y} + S_{1,z} S_{2,z})]. \end{aligned} \tag{2.44}$$

The introduction of the rotating frame suggests that it will be more convenient to use the spin raising and lowering operators $S_+ = S_x + iS_y$ and $S_- = S_x - iS_y$, respectively. Inverting these expressions for S_x and S_y we find

$$\begin{aligned} S_x &= \frac{1}{2}(S_+ + S_-) \\ S_y &= \frac{1}{2i}(S_+ - S_-), \end{aligned} \tag{2.45}$$

so that

$$\begin{aligned}
S_x S_x &= \frac{1}{4}(S_+ + S_-)(S_+ + S_-) \\
&= \frac{1}{4}(S_+ S_+ + S_+ S_- + S_- S_+ + S_- S_-) \\
S_x S_y &= \frac{1}{4i}(S_+ + S_-)(S_+ - S_-) \\
&= \frac{1}{4i}(S_+ S_+ - S_+ S_- + S_- S_+ - S_- S_-) \\
S_y S_x &= \frac{1}{4i}(S_+ - S_-)(S_+ + S_-) \\
&= \frac{1}{4i}(S_+ S_+ + S_+ S_- - S_- S_+ - S_- S_-) \\
S_y S_y &= -\frac{1}{4}(S_+ - S_-)(S_+ - S_-) \\
&= -\frac{1}{4}(S_+ S_+ - S_+ S_- - S_- S_+ + S_- S_-).
\end{aligned} \tag{2.46}$$

In the rotating frame, each term picks up a rotation term with sign opposite to that of the raising (+) or lowering (-) operator it is evolving and oscillates at the Larmor frequency. The time evolution generated by $H_0 = -\omega_L S_z$ is $\tilde{S}_+(t) = e^{-i\omega_L t} S_+$, $S_-(t) = e^{+i\omega_L t} S_-$, and $\tilde{S}_z(t) = S_z$ since, for example, transforming S^+ to the rotating frame gives

$$e^{+i\omega_L M^z} S^+ e^{-i\omega_L M^z} \rightarrow e^{-i\omega_L t} S^+. \tag{2.47}$$

The previous transformation shows that S^+ for the i th spin commutes with the S_z operator for all other spins, and it also reveals that the transformation amounts to nothing more than a phase shift for the S^+ (and also S^-) operator.

The terms in the rotating frame are

$$\begin{aligned}
S_x S_x &\rightarrow \frac{1}{4}(S_+ S_+ e^{-2i\omega_L t} + S_+ S_- + S_- S_+ + S_- S_- e^{+2i\omega_L t}) \\
S_x S_y &\rightarrow \frac{1}{4i}(S_+ S_+ e^{-2i\omega_L t} - S_+ S_- + S_- S_+ - S_- S_- e^{+2i\omega_L t}) \\
S_y S_x &\rightarrow \frac{1}{4i}(S_+ S_+ e^{-2i\omega_L t} + S_+ S_- - S_- S_+ - S_- S_- e^{+2i\omega_L t}) \\
S_y S_y &\rightarrow -\frac{1}{4}(S_+ S_+ e^{-2i\omega_L t} - S_+ S_- - S_- S_+ + S_- S_- e^{+2i\omega_L t}).
\end{aligned} \tag{2.48}$$

With the assumption that the contribution from the quickly oscillating terms will be negligible, we

find

$$\begin{aligned}
S_x S_x &\rightarrow \frac{1}{4}(S_+ S_- + S_- S_+) \\
S_x S_y &\rightarrow \frac{1}{4i}(-S_+ S_- + S_- S_+) \\
S_y S_x &\rightarrow \frac{1}{4i}(S_+ S_- - S_- S_+) \\
S_y S_y &\rightarrow \frac{1}{4}(S_+ S_- + S_- S_+).
\end{aligned} \tag{2.49}$$

Since the rotating frame spin operator is $\tilde{S}_z(t) = S_z$, the time evolution of terms involving S_z are given as

$$\begin{aligned}
S_z S_z &\rightarrow S_z S_z \\
S_z S_x &\rightarrow \frac{1}{2} S_z (S_+ e^{-i\omega_L t} + S_- e^{+i\omega_L t}) \rightarrow 0 \\
S_z S_y &\rightarrow \frac{1}{2i} S_z (S_+ e^{-i\omega_L t} - S_- e^{+i\omega_L t}) \rightarrow 0,
\end{aligned} \tag{2.50}$$

and the cross terms oscillate at the Larmor frequency which can then be dropped in the secular approximation. With these approximations in hand, the dipole-dipole interaction Hamiltonian given in Eq. 2.44 in Cartesian coordinates can be expressed as

$$\begin{aligned}
\frac{H'_{DD}}{-b_{i,j}} &\rightarrow S_z S_z (3 \cos^2 \theta - 1) \\
&+ \frac{1}{4} (S_+ S_- + S_- S_+) (3 \sin^2 \theta \cos^2 \phi - 1 + 3 \sin^2 \theta \sin^2 \phi - 1) \\
&+ \frac{1}{4i} (-S_+ S_- + S_- S_+) \sin^2 \theta \sin \phi \cos \phi \\
&+ \frac{1}{4i} (S_+ S_- - S_- S_+) \sin^2 \theta \sin \phi \cos \phi,
\end{aligned} \tag{2.51}$$

where the prime in H'_{DD} indicates that this is the secular approximation to the exact dipole-dipole interaction Hamiltonian. It is clear to see that the last two lines in the previous equation will cancel, and the Pythagorean relationship can be used to simplify the trigonometric expressions in the second line to produce

$$\frac{H'_{DD}}{-b_{i,j}} \rightarrow S_z S_z (3 \cos^2 \theta - 1) + \frac{1}{4} (S_+ S_- + S_- S_+) (3 \sin^2 \theta - 2). \tag{2.52}$$

The final factor in the last term of the previous equation can be rearranged as

$$\begin{aligned}
 & 3 \sin^2 \theta - 2 \\
 & 3(1 - \cos^2 \theta) - 2 \\
 & -3 \cos^2 \theta + 1 \\
 & 1 - 3 \cos^2 \theta.
 \end{aligned} \tag{2.53}$$

Substituting this into Eq. 2.52 gives

$$\begin{aligned}
 \frac{H'_{DD}}{-b_{i,j}} & \rightarrow S_z S_z (3 \cos^2 \theta - 1) + \frac{1}{4} (S_+ S_- + S_- S_+) (1 - 3 \cos^2 \theta) \\
 & \rightarrow (3 \cos^2 \theta - 1) [S_z S_z - \frac{1}{4} (S_+ S_- + S_- S_+)] \\
 & \rightarrow (3 \cos^2 \theta - 1) [S_z S_z - \frac{1}{2} (S_x S_x + S_y S_y)] \\
 & \rightarrow \left(\frac{3 \cos^2 \theta - 1}{2} \right) (S_x S_x + S_y S_y - 2 S_z S_z) \\
 & \rightarrow \left(\frac{3 \cos^2 \theta - 1}{2} \right) (\vec{S}_1 \cdot \vec{S}_2 - 3 S_z S_z).
 \end{aligned} \tag{2.54}$$

Finally, the secular approximation of the dipole-dipole interaction Hamiltonian can be expressed as

$$\boxed{H'_{DD} = b_{i,j} \left(\frac{3 \cos^2 \theta - 1}{2} \right) (3 S_{z,1} S_{z,2} - \vec{S}_1 \cdot \vec{S}_2)} \tag{2.55}$$

2.3.2 J-coupling interaction approximation

Starting with the expression for the exact J-coupling interaction Hamiltonian between two spins i and j in angular frequency units, we have

$$H_J = 2\pi \vec{S}_i \overleftrightarrow{J} \vec{S}_j \tag{2.56}$$

with the J-coupling tensor $\overleftrightarrow{\mathcal{J}}$ given by Eq. 2.34 and the vector spin operators $\vec{S}_{i,j}$ of the two spins involved in the interaction. Carrying out the matrix multiplication explicitly, we have

$$\begin{aligned} \vec{S}_i \overleftrightarrow{\mathcal{J}} \vec{S}_j = & \\ & J'_{xx} S_x S_x + J'_{xy} S_x S_y + J'_{xz} S_x S_z \\ & + J'_{yx} S_y S_x + J'_{yy} S_y S_y + J'_{yz} S_y S_z \\ & + J'_{zx} S_z S_x + J'_{zy} S_z S_y + J'_{zz} S_z S_z. \end{aligned} \quad (2.57)$$

Under time evolution, the terms which oscillate at the Larmor frequency or at twice the Larmor precession frequency can be omitted in the secular approximation. As seen in Eq. 2.50, the terms with exactly one S_z factor oscillate at the Larmor frequency and may therefore be omitted from the Hamiltonian. Equation 2.57 can then be rewritten as

$$\begin{aligned} \vec{S}_i \overleftrightarrow{\mathcal{J}} \vec{S}_j = & \\ & J'_{xx} S_x S_x + J'_{yy} S_y S_y + J'_{zz} S_z S_z \\ & + J'_{xy} S_x S_y + J'_{yx} S_y S_x. \end{aligned} \quad (2.58)$$

From Eqs. 2.49 and 2.50, we can substitute into Eq. 2.58 to find

$$\begin{aligned} \vec{S}_i \overleftrightarrow{\mathcal{J}} \vec{S}_j = & \\ & \frac{J'_{xx}}{4} (S_+ S_- + S_- S_+) + \frac{J'_{yy}}{4} (S_+ S_- + S_- S_+) + J'_{zz} S_z S_z \\ & + \frac{J'_{xy}}{4i} (-S_+ S_- + S_- S_+) + \frac{J'_{yx}}{4i} (S_+ S_- - S_- S_+). \end{aligned} \quad (2.59)$$

A simplification can be made here by noting that the tensor must be symmetric if it can be expressed as a diagonal matrix in the principal axis basis, so $J_{xy} = J_{yx}$. With this simplification Eq. 2.59 can be written as

$$\begin{aligned} \vec{S}_i \overleftrightarrow{\mathcal{J}} \vec{S}_j = & \frac{1}{4} (J'_{xx} + J'_{yy}) (S_+ S_- + S_- S_+) + J'_{zz} S_z S_z \\ = & \frac{1}{8} (J'_{xx} + J'_{yy}) 2(S_+ S_- + S_- S_+) + J'_{zz} S_z S_z \\ = & \frac{1}{8} (J'_{xx} + J'_{yy}) [(S_+ S_- + S_- S_+) + (S_+ S_- + S_- S_+)] + J'_{zz} S_z S_z \\ = & \boxed{\frac{1}{2} (J'_{xx} + J'_{yy}) (S_x S_x + S_y S_y) + J'_{zz} S_z S_z}. \end{aligned} \quad (2.60)$$

Thus, even in a general orientation, the J-coupling interaction in the secular approximation depends only on the diagonal tensor elements.

It is convenient to define two quantities which relate to observables in this project. The first is $J' \equiv \frac{1}{3}(J'_{xx} + J'_{yy} + J'_{zz})$, which relates to the isotropic part of the J-coupling. The second quantity is $\Delta J' \equiv J'_{zz} - \frac{1}{2}(J'_{xx} + J'_{yy})$ which relates to the anisotropic part of the J-coupling. In terms of these quantities Eq. 2.60 can be written as

$$\begin{aligned}
\vec{S}_i \overset{\leftrightarrow}{J} \vec{S}_j &= \frac{1}{2}(J'_{xx} + J'_{yy})(S_x S_x + S_y S_y) + J'_{zz} S_z S_z \\
&= \frac{1}{2}(J'_{xx} + J'_{yy} + J'_{zz})(S_x S_x + S_y S_y) + J'_{zz}[S_z S_z - \frac{1}{2}(S_x S_x + S_y S_y)] \\
&= \frac{1}{2}(J'_{xx} + J'_{yy} + J'_{zz})(S_x S_x + S_y S_y + S_z S_z) \\
&\quad + J'_{zz}[S_z S_z - \frac{1}{2}(S_x S_x + S_y S_y)] - \frac{1}{2}(J'_{xx} + J'_{yy} + J'_{zz})S_z S_z \\
&= \frac{3}{2}J'\vec{S}_1 \cdot \vec{S}_2 + \frac{1}{2}(J'_{zz} - J'_{xx} - J'_{yy})S_z S_z - \frac{1}{2}J'_{zz}(S_x S_x + S_y S_y) \\
&= \frac{3}{2}J'\vec{S}_1 \cdot \vec{S}_2 + [J'_{zz} - \frac{1}{2}(J'_{xx} + J'_{yy})]S_z S_z - \frac{1}{2}J'_{zz}\vec{S}_1 \cdot \vec{S}_2 \\
&= \frac{3}{2}J'\vec{S}_1 \cdot \vec{S}_2 + \Delta J' S_z S_z - \frac{1}{2}J'_{zz}\vec{S}_1 \cdot \vec{S}_2 \\
&= \frac{3}{2} \left[\frac{1}{3}(J'_{xx} + J'_{yy} + J'_{zz}) - \frac{2}{3} \left(\frac{1}{2}J'_{zz} \right) \right] \vec{S}_1 \cdot \vec{S}_2 + \Delta J' S_z S_z \\
&= \frac{3}{2} \left[\frac{1}{3}(J'_{xx} + J'_{yy} + J'_{zz}) - \frac{1}{3}J'_{zz} \right] \vec{S}_1 \cdot \vec{S}_2 + \Delta J' S_z S_z \\
&= \left[\frac{1}{3}(J'_{xx} + J'_{yy} + J'_{zz}) + \frac{1}{6}(J'_{xx} + J'_{yy} + J'_{zz}) - \frac{1}{2}J'_{zz} \right] \vec{S}_1 \cdot \vec{S}_2 + \Delta J' S_z S_z \\
&= \left[\frac{1}{3}(J'_{xx} + J'_{yy} + J'_{zz}) - \frac{1}{3}J'_{zz} + \frac{1}{6}(J'_{xx} + J'_{yy}) \right] \vec{S}_1 \cdot \vec{S}_2 + \Delta J' S_z S_z \\
&= \left(J' - \frac{1}{3}\Delta J' \right) \vec{S}_1 \cdot \vec{S}_2 + \Delta J' S_z S_z.
\end{aligned} \tag{2.61}$$

To this point the derivation has been for a J-coupling tensor in any orientation. Next we relate J' and $\Delta J'$ in terms of components of $\overset{\leftrightarrow}{J}$ in any orientation to J and ΔJ from Eq. 2.36 which are defined in the principal axis coordinate basis.

If we further restrict the J-coupling tensor to reflect the specific symmetries found in the single-crystal silicon used in this project, we find the tensor to have the same structure as in Eq. 2.35 with the first two diagonal entries being equal and the third different. A general rotation can

be specified by the three Euler angles α , β , and γ which specify rotations about a fixed z-axis, y-axis, and then z-axis again, respectively. In this convention, the angle β specifies the rotation angle measured from the z-axis in the principal axis coordinate basis. The general rotation matrix can be constructed by multiplying the three separate rotations about their respective fixed axes in reverse order as

$$\begin{aligned}
 R_x(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \\
 R_y(\phi) &= \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} \\
 R_z(\psi) &= \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 R &= R_z(\gamma)R_y(\beta)R_x(\alpha).
 \end{aligned} \tag{2.62}$$

Using a shorthand notation for the sine and cosine functions as $\sin(x) \rightarrow s(x)$ and $\cos(x) \rightarrow c(x)$, the rotation matrix R can be specified as

$$R(\alpha, \beta, \gamma) = \begin{bmatrix} c(\alpha)c(\beta)c(\gamma) - s(\alpha)s(\gamma) & -c(\alpha)c(\beta)s(\gamma) - s(\alpha)c(\gamma) & c(\alpha)s(\beta) \\ s(\alpha)c(\beta)c(\gamma) + c(\alpha)s(\gamma) & -s(\alpha)c(\beta)s(\gamma) + c(\alpha)c(\gamma) & s(\alpha)s(\beta) \\ -s(\beta)c(\gamma) & s(\beta)s(\gamma) & c(\beta). \end{bmatrix} \tag{2.63}$$

When the axially symmetric J-coupling tensor represented in the principal axis coordinate basis is rotated by $\overleftrightarrow{J}' = R^T \overleftrightarrow{J} R$ into the new basis, the new values for the tensor elements are represented

as

$$\begin{aligned}
\overleftrightarrow{J}' &= R^T \overleftrightarrow{J} R = \\
J'_{xx} &= \frac{1}{2}c^2(\alpha)(J_{xx} + J_{zz} + (J_{xx} - J_{zz})c(2\beta)) + J_{xx}s^2(\alpha) \\
J'_{xy} &= -(J_{xx} - J_{zz})c(\alpha)s(\alpha)s^2(\beta) \\
J'_{xz} &= (-J_{xx} + J_{zz})c(\alpha)c(\beta)s(\beta) \\
J'_{yx} &= -(J_{xx} - J_{zz})c(\alpha)s(\alpha)s^2(\beta) \\
J'_{yy} &= J_{xx}c^2(\alpha) + \frac{1}{2}(J_{xx} + J_{zz} + (J_{xx} - J_{zz})c(2\beta))s^2(\alpha) \\
J'_{yz} &= (-J_{xx} + J_{zz})c(\beta)s(\alpha)s(\beta) \\
J'_{zx} &= (-J_{xx} + J_{zz})c(\alpha)c(\beta)s(\beta) \\
J'_{zy} &= (-J_{xx} + J_{zz})c(\beta)s(\alpha)s(\beta) \\
J'_{zz} &= \frac{1}{2}(J_{xx} + J_{zz} + (-J_{xx} + J_{zz})c(2\beta)).
\end{aligned} \tag{2.64}$$

The value of $J' = \frac{1}{3}(J'_{xx} + J'_{yy} + J'_{zz})$ can be found for this new tensor as

$$\begin{aligned}
J' &= \frac{1}{3} \left[\frac{1}{2}c^2(\alpha)(J_{xx} + J_{zz} + (J_{xx} - J_{zz})c(2\beta)) + J_{xx}s^2(\alpha) \right. \\
&\quad + J_{xx}c^2(\alpha) + \frac{1}{2}(J_{xx} + J_{zz} + (J_{xx} - J_{zz})c(2\beta))s^2(\alpha) \\
&\quad \left. + \frac{1}{2}(J_{xx} + J_{zz} + (-J_{xx} + J_{zz})c(2\beta)) \right] \\
&= \frac{1}{3} \left[\frac{1}{2}(J_{xx} + J_{zz} + (J_{xx} - J_{zz})c(2\beta)) \right. \\
&\quad \left. + J_{xx} + \frac{1}{2}(J_{xx} + J_{zz} + (-J_{xx} + J_{zz})c(2\beta)) \right] \\
&= \frac{1}{3}(2J_{xx} + J_{zz}) \\
&= J
\end{aligned} \tag{2.65}$$

The value of $\Delta J' = J'_{zz} - \frac{1}{2}(J'_{xx} + J'_{yy})$ can be found for this new tensor as

$$\begin{aligned}
\Delta J' &= \frac{1}{2}(J_{xx} + J_{zz} + (-J_{xx} + J_{zz})c(2\beta)) \\
&\quad - \frac{1}{2} \left[\frac{1}{2}c^2(\alpha)(J_{xx} + J_{zz} + (J_{xx} - J_{zz})c(2\beta)) + J_{xx}s^2(\alpha) \right. \\
&\quad \left. + J_{xx}c^2(\alpha) + \frac{1}{2}(J_{xx} + J_{zz} + (J_{xx} - J_{zz})c(2\beta))s^2(\alpha) \right] \\
&= \frac{1}{2}(J_{xx} + J_{zz} + (-J_{xx} + J_{zz})c(2\beta)) \\
&\quad - \frac{1}{2} \left[\frac{1}{2}(J_{xx} + J_{zz} + (J_{xx} - J_{zz})c(2\beta)) + J_{xx} \right] \\
&= \frac{1}{2}(J_{xx} + J_{zz} + (-J_{xx} + J_{zz})c(2\beta)) \\
&\quad - \frac{3}{4}(J_{xx}) - \frac{1}{4}(J_{zz} + (J_{xx} - J_{zz})c(2\beta)) \\
&= -\frac{1}{4}(J_{xx}) + \frac{1}{4}(J_{zz}) - \frac{3}{4}(J_{xx} - J_{zz})c(2\beta) \\
&= -\frac{1}{4}(J_{xx} - J_{zz}) - \frac{3}{4}(J_{xx} - J_{zz})c(2\beta) \\
&= -\frac{1}{4}(J_{xx} - J_{zz})(1 + 3c(2\beta)) \\
&= \frac{1}{4}\Delta J(1 + 3c(2\beta)) \\
&= \frac{1}{4}\Delta J(1 + 3(2c^2(\beta) - 1)) \\
&= \frac{1}{4}\Delta J(1 + 6c^2(\beta) - 3) \\
&= \frac{1}{4}\Delta J(6c^2(\beta) - 2) \\
&= \Delta J \left(\frac{3 \cos^2 \beta - 1}{2} \right).
\end{aligned} \tag{2.66}$$

So then the expression for the J-coupling interaction Hamiltonian can be given for an arbitrary orientation by

$$\begin{aligned}
\vec{S}_i \overset{\leftarrow}{J'} \vec{S}_j &= J' \vec{S}_i \cdot \vec{S}_j - \frac{1}{3}\Delta J' \vec{S}_i \cdot \vec{S}_j + \Delta J' S_z S_z \\
&= J \vec{S}_i \cdot \vec{S}_j + \frac{\Delta J}{3} \left(\frac{3 \cos^2 \beta - 1}{2} \right) (3S_z S_z - \vec{S}_i \cdot \vec{S}_j)
\end{aligned} \tag{2.67}$$

Therefore, the expression for the J-coupling interaction Hamiltonian in a general tensor basis

after the application of the secular approximation can be written as

$$H'_J = \sum_{i < j} 2\pi \left[J \vec{S}_i \cdot \vec{S}_j + \frac{\Delta J}{3} \left(\frac{3 \cos^2 \beta - 1}{2} \right) (3S_z S_z - \vec{S}_i \cdot \vec{S}_j) \right]. \quad (2.68)$$

Finally, the full secular approximation Hamiltonian can be written as

$$\boxed{\begin{aligned} H' = & \sum_{i < j} b_{i,j} \left(\frac{3 \cos \theta_{i,j} - 1}{2} \right) (\vec{S}_i \cdot \vec{S}_j - 3S_i^z S_j^z) \\ & - \sum'_{i < j} 2\pi \frac{\Delta J}{3} \left(\frac{3 \cos \theta_{i,j} - 1}{2} \right) (\vec{S}_i \cdot \vec{S}_j - 3S_i^z S_j^z) \\ & + \sum'_{i < j} 2\pi J \vec{S}_i \cdot \vec{S}_j, \end{aligned}} \quad (2.69)$$

where the primes on the last two summations indicate the sums are to be taken over nearest neighbors only. The angles $\theta_{i,j}$ represent the angles between each of the bond axes and the direction of the externally applied magnetic field.

2.4 Quantum spin dynamics

2.4.1 The initial state of the system

Since the Zeeman interaction is the strongest of the three interactions, we may say that it is the only interaction contributing to the initial thermal equilibrium state given by Eq. 2.8. From Eq. 2.19, the initial state is proportional to the magnetization

$$M^z = \sum_i S_i^z. \quad (2.70)$$

or the initial density matrix is

$$\rho(t=0) \propto M^z. \quad (2.71)$$

This is called the deviation density operator for the system because we omit the term in Eq. 2.14 proportional to the identity operator.

2.4.2 Relaxation and the time constants T_1 and T_2

In this experiment the observable is the transverse magnetization \hat{M}^x . The expectation value of the magnetization is

$$\langle M^x \rangle = \text{Tr}(\tilde{M}^x \tilde{\rho}). \quad (2.72)$$

In a standard pulse-acquire NMR experiment, the sample is placed in the external field and allowed to thermalize to the initial state, which was derived above to be $\rho_0 = M^z$. The time constant T_1 is associated with the sample magnetization M^z . For silicon the T_1 time is about 4 hours (see Sec. 5.2 for details). This quantity is related to the strength of the interaction of the spins with the thermal bath of lattice vibrations.

During the experiment, a transverse radio frequency (RF) signal is applied which has the effect of rotating the spin vectors of the active nuclei about the transverse axis by an amount proportional to the strength and duration of the applied RF field. The RF signal is shut off, and the same coils are then used to detect the transverse magnetization of the precessing spins in the active nuclei. This spin precession is about the external field axis (assumed to be the \hat{z} direction), and the transverse magnetization signal is recorded by the RF coils around the sample. As the spins precess, the dipole interactions between spins cause the observable magnetization to decay.. This manifests as a decay of the signal recorded by the RF coils. Because of this decaying behavior, this signal is known as the free-induction decay (FID) signal.

This oscillating signal can be Fourier transformed to produce the frequency spectrum which is the characteristic line shape of NMR spectroscopy. The time constant T_2 of the signal decay is associated with the strength of the interactions between active nuclei. In simplified models, the FID is modeled as a decaying exponential function. In the rotating frame, this can be expressed as $FID(t) = \langle M^x \rangle \propto e^{-t/T_2}$. The Fourier transform of a decaying exponential with decay constant $1/T_2$ is a Lorentzian with the full-width at half maximum given as $\frac{\pi}{T_2}$. The T_1 time constant is sometimes much longer than T_2 for spin-1/2 nuclei in hard solids, as is the case in this project.

Chapter 3

Materials

This project is concerned with understanding the many-body spin dynamics of a macroscopic collection of interacting spin-1/2 nuclei. Since we seek to understand the interactions of a disordered system, it is necessary to characterize this disorder. The atoms in a crystal lattice are ordered by periodicity of the lattice, but they are disordered by random occupation at the natural isotopic abundance. The simplest such system to study is a material with only isotopes which are either spin-0 or spin-1/2.

Crystals made from small organic acids, such as oxalic and malonic acid, were also originally considered for use in this project because of the relative ease by which isotopically purified samples of ^{13}C enriched molecules can be obtained commercially. However, the protons in these acids form an ordered lattice at nearly 100% natural abundance and are strongly coupled to the thermal bath as well as the carbon atoms in an organic lattice. This reduces the T_1 time and coherence of the ^{13}C nuclei and makes these systems less attractive. Silicon was then proposed as one of the next most interesting candidates for a disordered coherent quantum system, with the potential for further studies involving isotopically enriched samples to manipulate the disorder parameter.

3.1 Why silicon?

There are several qualities which make single-crystal silicon a suitable candidate material for the purpose of this project. Recall that nuclei with spin $> 1/2$ are not desirable because the electric quadrupole interaction of such nuclei greatly reduces the T_1 time.

There are many elements on the periodic table which have stable spin-1/2 nuclei, including ^{13}C and ^{29}Si . Spin > 0 is necessary to create spin-spin interactions in the first place. However, at natural isotopic abundance some elements which have stable spin-1/2 isotopes also have higher spin isotopes present. Note that carbon and silicon do not have any stable isotopes with spin greater than 1/2, which therefore make them excellent candidate materials.

Another attractive feature of both silicon and diamond is that they both belong to the same crystal class which has no magnetically inequivalent sites. This means that the expected NMR line shape for the material will only have one primary peak as well as other smaller orientation dependent splittings. It is also worth noting explicitly that there are no protons in single-crystal silicon and diamond. As described above, these protons would reduce T_1 and coherence.

There are compounds which are composed of only silicon or carbon and other nearly magnetically inert atoms (spin-0), such as pure silicon, diamond, and the SiO_2 α -phase crystal structure known as quartz. Quartz has magnetically inequivalent sites which will produce multiple peaks in the transform of the FID. These peaks can be manipulated individually using NMR pulse sequences designed to interact with only one or more of the peaks at a time. This peak manipulation feature can be used for future experiments to understand spin diffusion in the macroscopic sample with potential relevance to many-body localization. Because the chemical shift anisotropy is small in quartz, there is no need for high precision angular alignment in the sample, as would be the case with other materials with large chemical shift anisotropies. This feature, as well as the magnetically inequivalent atomic sites in the quartz lattice, make it an excellent candidate material.

3.2 Properties of Silicon

Single-crystal silicon belongs to the same crystal class as diamond where every atom in the lattice is bonded to four other atoms in a regular tetrahedral arrangement. The bond length between nearest neighbors is $a\sqrt{3}/4 \approx 2.35 \text{ \AA}$, and there are eight total atoms in the conventional unit cell. See Figs. 3.1 and 3.2 for depictions of the silicon conventional and primitive unit cells, respectively.

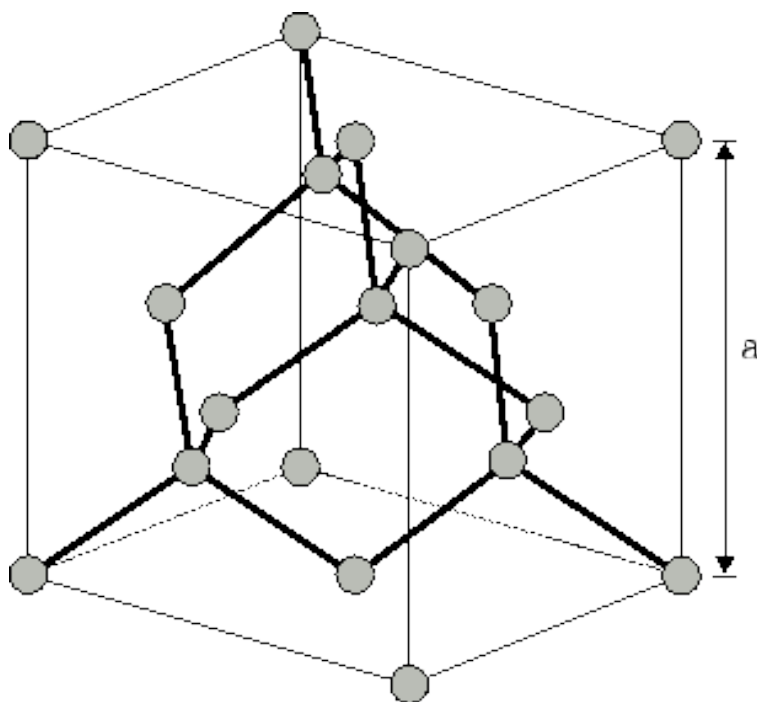


Figure 3.1: The silicon conventional unit cell. Similar to the diamond lattice there are eight atoms in the basis, and the lattice constant is $a = 5.43096 \text{ \AA}$. Image reproduced from <http://www.iue.tuwien.ac.at/phd/ungersboeck/node27.html>

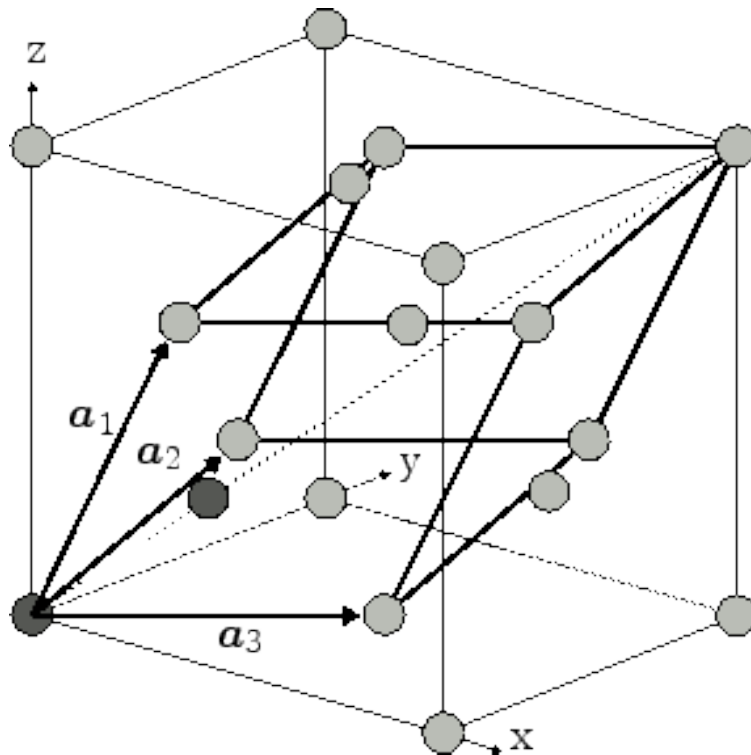


Figure 3.2: The silicon primitive unit cell. There are two atoms in the basis, one at the origin and the other at a quarter of the length along the body diagonal of the conventional unit cell. The primitive lattice translation vectors are the vectors which connect the atom at the origin to one of the atoms at the center of one of the faces which intersects the origin. Image reproduced from <http://www.iue.tuwien.ac.at/phd/ungersboeck/node27.html>

To understand the angular dependence of the dipole-dipole interactions exhibited in the NMR line shape, four different orientations of the silicon lattice with respect to the externally applied magnetic field are considered. These orientations are labeled in Fig. 3.3 in Miller index notation.

The samples which were prepared for this project are made from float-zone silicon wafers with specified resistivity of 5 k Ω cm to 10 k Ω cm.

3.3 Susceptibility matching

In this project we seek to understand the physical principals of solid state NMR spectroscopy for single-crystal silicon to eventually obtain experimental insight into many-body localization. To obtain the narrowest possible line shape from the spectrometer, a uniform magnetic field throughout the sample is required. In liquid samples this is usually not an issue since cylindrical geometries of homogeneous magnetic susceptibility produce uniform magnetization in the material in the presence of an externally applied magnetic field, and NMR test tubes exhibit a cylindrical geometry.

In our particular project, the material sample is a stack of smooth flat rectangular silicon chips approximating a rectangular prism with epoxy to affix each layer in place relative to one another. The geometry is not cylindrical, which introduces line broadening in the NMR line shape. This broadening obscures the information about the spin-spin couplings which are of primary interest in this experiment, and so measures must be taken to correct this field distortion.

If the silicon sample is surrounded by a solution with a similar magnetic susceptibility in a cylindrical NMR test tube, then the field distortions caused by susceptibility inhomogeneity would be minimal compared to previous experiments. In previous experiments involving solid-state NMR spectroscopy, no reported efforts were taken to reduce line broadening caused by inhomogeneity in the magnetic field [5, 6, 7, 8, 9, 10]. This inhomogeneity is caused by a magnetic susceptibility mismatch between the crystal sample and the surrounding environment. The shim coils surrounding RF coils in the spectrometer provide a method for correcting inhomogeneities present in the applied field.

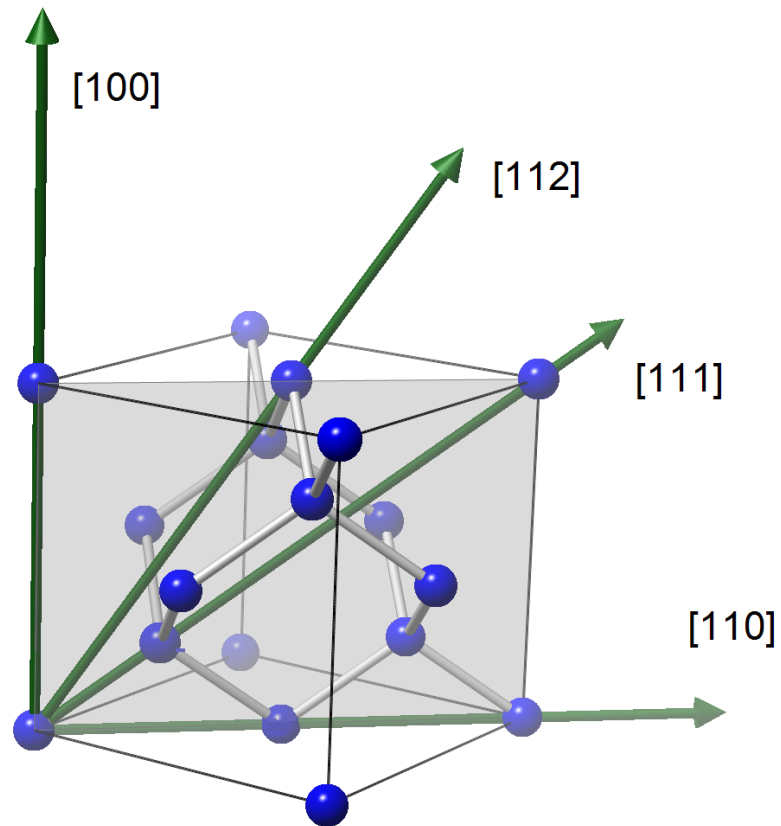


Figure 3.3: The conventional silicon unit cell and four experimental orientations. The four orientations of the externally applied magnetic field relative to the unit cell which are explored in this project. Reproduced from Ref. [1].

The magnetization vector of a material is given by

$$\vec{M} = \chi_v \vec{H}, \quad (3.1)$$

where \vec{M} is the induced magnetization of the material or magnetic dipole moment per unit volume (measured in Amperes per meter in SI units), χ_v is the volume magnetic susceptibility of the material (a constant dimensionless quantity), and \vec{H} is the magnetic field strength (also in units of Amperes per meter in SI units). The relationship between the magnetic field \vec{H} and the magnetic flux density \vec{B} is

$$\vec{B} = \mu_0(\vec{H} + \vec{M}) = \mu_0(1 + \chi_v)\vec{H} = \mu\vec{H}, \quad (3.2)$$

which indicates that the flux density is proportional to the sum of the magnetic field and the magnetization in a sample. For more on magnetic susceptibility, see Ref. [21].

A mismatch in magnetic susceptibility causes distortions in the applied field. A uniform magnetic field will produce the smallest possible line width, so any inhomogeneity in the applied field will cause line broadening in the transform of the FID signal. In a susceptibility matched sample, the applied field will be uniform, and therefore produce the least amount of line broadening. When the broadening is reduced, fine features of the signal may be resolved. This is important in this project since the dipole-dipole interaction produces features spread over a small region of about 100 Hz, or 1 ppm at the 100 MHz Larmor frequency of ^{29}Si .

For single-crystal silicon $\chi_{v,\text{Si}} = -3.4 \times 10^{-6}$ in SI units, where the negative sign indicates that silicon is a diamagnetic material. The magnetic susceptibility of air is $\chi_{v,\text{air}} = 3.6 \times 10^{-7}$, indicating that air is slightly paramagnetic. This shows the size of the mismatch in χ_v between the two materials to be $\Delta\chi = 3.76 \times 10^{-6}$.

Since single-crystal silicon is a diamagnetic material, a diamagnetic material is also needed to surround the silicon sample to attempt to match its magnetic susceptibility. The magnetic susceptibility of acetone is $\chi_{v,\text{acetone}} = -5.8 \times 10^{-6}$, which is close to that of silicon and even more diamagnetic. To increase the susceptibility of the diamagnetic solvent, a paramagnetic dopant must be used. Too much paramagnetic dopant could interfere with the experiment, which is why

it is important that the undoped solvent be more diamagnetic than the silicon sample by a small amount relative to the size of the natural susceptibility mismatch. The paramagnetic dopant used in this experiment is chromium (III) acetylacetonate or $\text{Cr}(\text{acac})_3$ with $\chi_{v,\text{Cr}(\text{acac})_3} > 0$ since it is paramagnetic.

The susceptibility matched solution contains protons in the acetone which may be used to shim the sample in the spectrometer. (See App. A for details concerning the shimming process.) The spectrometer used in this project is capable of switching between a proton signal and a silicon signal in the sample. This feature is useful for shimming the silicon signal by way of shimming to the proton signal and then switching channels. This is an operational convenience since the T_1 relaxation in the protons is on the order of seconds where the T_1 relaxation in the silicon is on the order of hours (see Sec. 5.2 for details).

The method for obtaining a susceptibility match between the sample and the surrounding solution is by direct measurement and adjustment *in situ*. A known quantity of the paramagnetic dopant is added to some amount of solvent which is known to produce a solution with volume magnetic susceptibility greater than that of the silicon (too paramagnetic). A basic pulse-acquire experiment is conducted on the protons in the solvent to determine the closeness of the susceptibility match. The line shape of the transformed FID signal indicates the closeness of this match. See Sec. 3.3.1 for modeling and details of the proton line shape. If the susceptibility of the sample and solution are exactly matched, then the line shape should exhibit a single peak with no distortions. If there is a mismatch in the susceptibilities, then there will appear to be a peak off to one side of the primary proton signal peak proportional to the strength of the mismatch and there will be characteristic distortions in the primary peak. The separation between the side peak and the central peak is directly proportional to the strength of the susceptibility mismatch.

The initial measurement is found to produce a line shape with features indicating a mismatch (by design the solution is initially too paramagnetic compared to the silicon). The separation between the side peak and central peak is recorded in Hz. The sample tube is then removed from the spectrometer, and a small volume of the solvent is added to lower the magnetic susceptibility

of the solution. The volume of this additional amount of solvent added to the solution is recorded. The sample tube is then placed back into the spectrometer, and another pulse-acquire experiment is performed. The separation between the side peak and the central peak in the line shape is again recorded in Hz.

A plot can now be made to relate the peak separation (in Hz) to the volume of added solvent. Since the peak separation is directly proportional to the volume of added solvent, the plot will have a linear behavior with volume of added solvent as the independent variable and the peak separation as the dependent variable. The two points defined by the two initial measurements can be used to construct a line. The intercept with the horizontal axis represents the point where the peak separation is zero for that volume of solvent added. When this volume of solvent is added to the solution, the susceptibility of the solution will, in principle, match the susceptibility of the silicon in the sample. This can be verified by running a pulse-acquire experiment on the protons in the solution. If the proton line shape resulting from the measurement is a single undistorted peak, then the susceptibility between the sample and the solution is matched. If the spectrometer is properly shimmed on the proton signal and then switched to the silicon channel, then the silicon channel is also shimmed. This ensures the smallest possible line width in the peak.

3.3.1 Modeling and testing the susceptibility mismatch

The susceptibility mismatch can be modeled using a Mathematica package called Radia [22, 23], which is designed for simulating magnetostatics systems from specified boundary conditions. (See the attached code in App. D.)

The geometry of the experiment is plotted with the magnetization at the sample points colored by their respective magnetization induced in the material by the externally applied magnetic field in Fig. 3.4. The susceptibility mismatch is initially $\Delta\chi_i = \chi_{\text{Silicon}} - \chi_{\text{Solution}} = -0.92 \times 10^{-6}$, and the final susceptibility mismatch is given by $\Delta\chi_j = \chi_{\text{Silicon}} - \chi_{\text{Solution}} = -0.10 \times 10^{-6}$.

The effect of a susceptibility mismatch between the two volumes in the geometry in the NMR line shape of the sample is shown in Fig. 3.5.

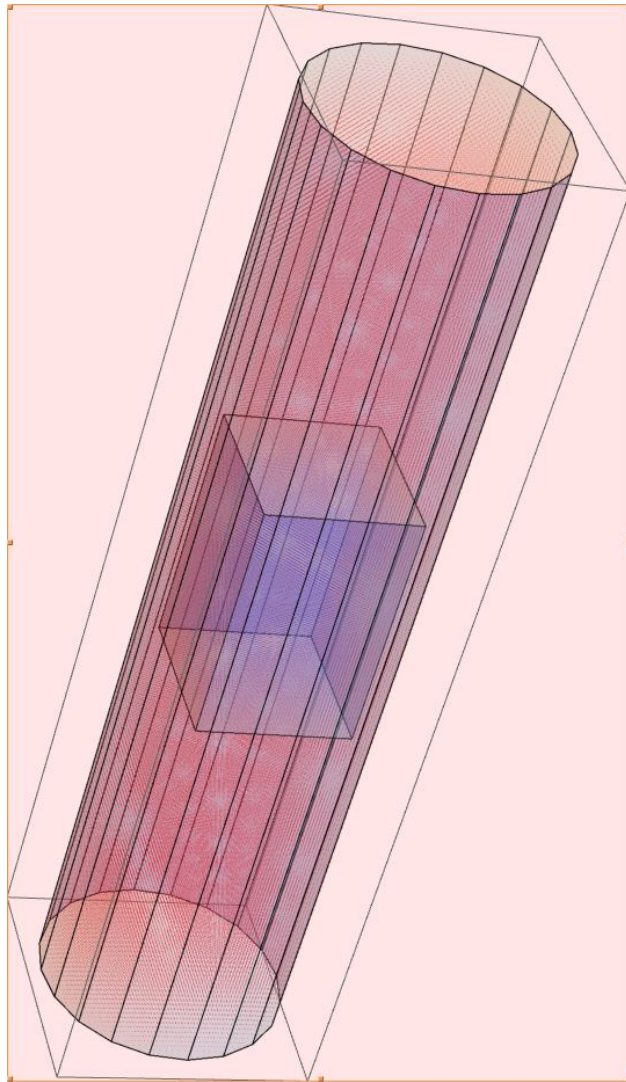


Figure 3.4: Radia model of the experimental sample geometry. Model generated by Mathematica package Radia showing the magnetization in the rectangular prism (blue) and the surrounding solution (red). The silicon sample is assumed to be less diamagnetic than the solution. The value of the magnetization induced by the externally applied magnetic field is evaluated at each of the points in the sample given the boundary value conditions. The solid is assumed to be less diamagnetic than the liquid, which is the usual case.

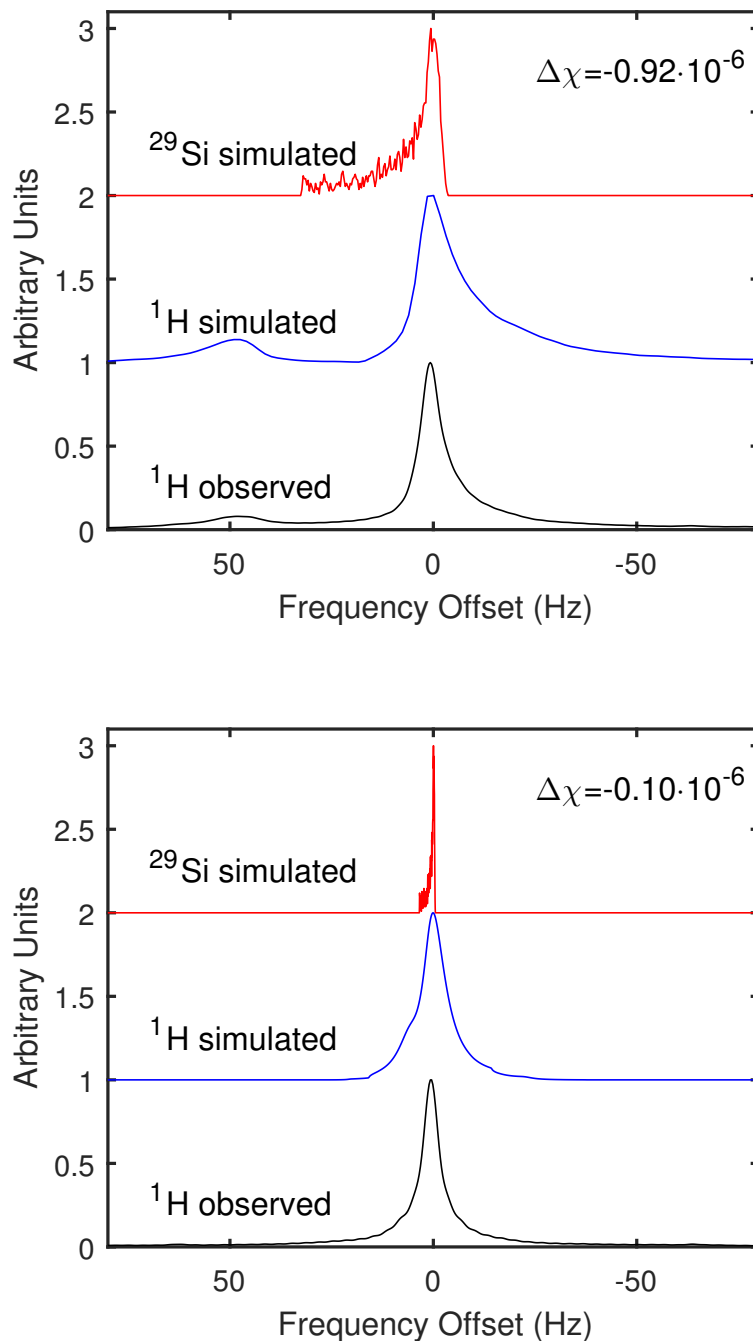


Figure 3.5: Radia model of proton line shape of susceptibility mismatch. Simulated line shapes for the contributions from ^{29}Si and ^1H separately plotted offset above the experimentally observed ^1H line shape for two different values of susceptibility mismatch $\Delta\chi$. As the susceptibility match improves, the line shapes are both narrower and less distorted. The proton line shapes are used to shim the externally applied field. When the external field is properly shimmed and the susceptibility of the sample is sufficiently uniform, the line shape produced in experiment will be as narrow as possible.

The method for testing and correcting the susceptibility mismatch in this project as described above has been used to obtain line shapes exhibiting the characteristic features also shown in Fig. 3.5, namely:

- (1) A side peak with much smaller amplitude compared to the central peak, which is offset by some frequency Δf and
- (2) A distortion in the shape of the central peak on the same side as the side peak.

3.4 Sample construction

Samples used in this project are all assembled using the same basic procedure. A typical sample is shown in Fig. 3.6. Before the crystal wafers are diced to correspond to a certain orientation they are affixed to a substrate with wax. The wafers are then diced into chips of size $2.6 \text{ mm} \times 5.0 \text{ mm} \times 0.525 \text{ mm}$. To remove the chips from the wax, a warm ($\approx 100^\circ\text{C}$) acetone bath is used. A second stage of acetone rinse is useful to remove the remaining wax from the freed chips.

The final assembly requires five chips and a capillary rod cut to 4.8 cm to fit inside the NMR tube. A small amount of epoxy is thoroughly mixed and a small amount ($< \text{one drop}$) is applied to one side of four of the chips. The only epoxy-free chip is then lifted with tweezers onto one of the other epoxied chips and is loosely positioned in place. Each of the other chips is similarly manipulated until all five chips are attached. Glass microscope slides are used to ensure that the sample is as close to a perfect rectangular parallelepiped as possible. The epoxy is allowed to fully set before the capillary is attached. The epoxy mixture is saved to ensure that it cannot be etched with a fingernail upon hardening. This hardness indicates that the chips are properly bonded to each other and will not come apart in the acetone solution. Note that if the mixing process is not thorough enough, then the epoxy will never fully harden.

The capillaries are cut to about 4.8 cm with the help of an etching tool to more cleanly make the cut. Using a metric ruler and a new batch of well-mixed epoxy, a small amount of epoxy (about one drop) is applied to one of the sides of the chip stack and the capillary is set on top with equal

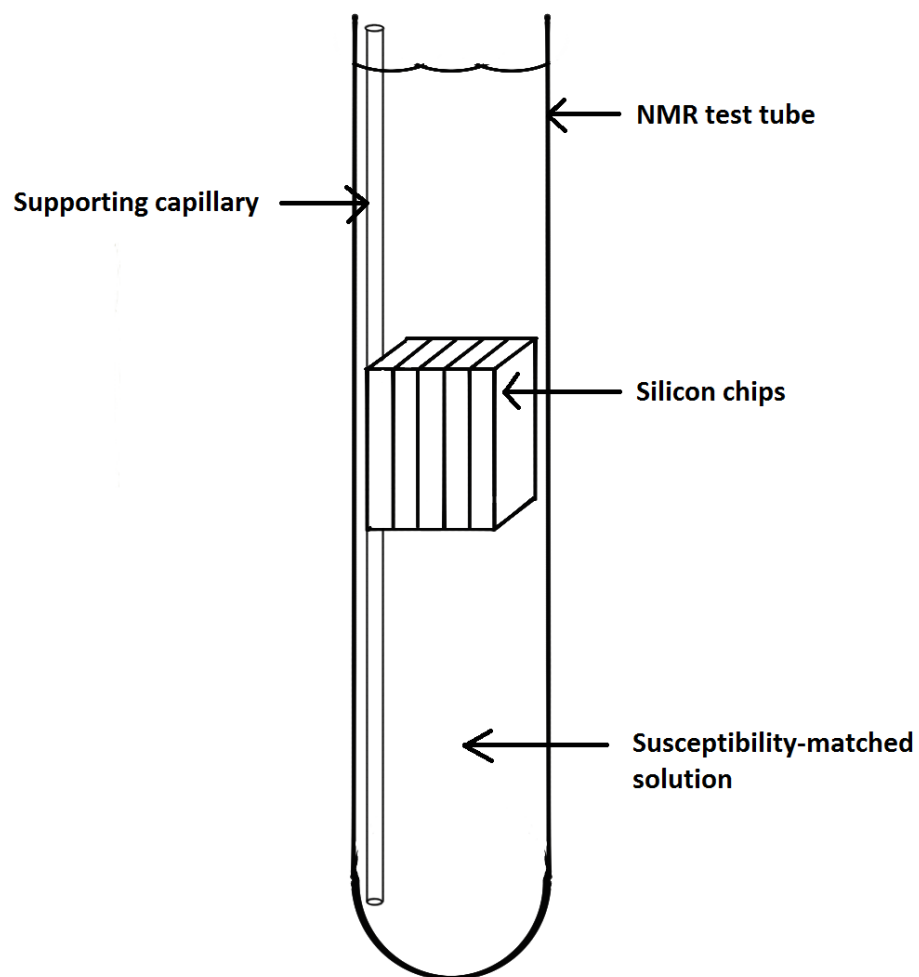


Figure 3.6: Diagram of constructed sample. The epoxied stack of silicon chips is attached to the support capillary with epoxy. This assembly is placed at the bottom of an NMR test tube and is surrounded by the susceptibility matched solution.

lengths on either side (i.e., about 2.4 cm on either side from the center of the sample). Again, enough time must be allowed to ensure that the epoxy is fully hardened.

To finish the sample preparation, the susceptibility-matched solvent solution is added to an empty NMR test tube and the sample on the capillary is inserted until the sample is fully immersed in the solution and the bottom of the capillary is flush with the bottom of the test tube. Care must be taken when the sample is inserted into the solution because bubbles can become trapped near the surface of the sample during immersion, which will cause distortion effects from susceptibility inhomogeneity as mentioned in the previous section.

3.5 X-ray diffractometry

The crystal structure and orientation of the sample can be verified using the technique of x-ray diffractometry. (For the specifics of diffractometer operation in this project, see App. B). One of the sample chips is mounted on a rotating arm which is placed in the path of a monochromatic x-ray beam. A CCD camera is located opposite the beam source to detect any x-rays which are deflected by Bragg diffraction.

The sample is rotated about a fixed axis perpendicular to the x-ray beam while the CCD collects information on the location and intensity of scattered x-rays. These data (collectively known as a diffraction pattern) are compared to the known diffraction patterns for the material in question to verify the orientation of the silicon crystals. For this project, the silicon sample diffraction pattern was compared to diffraction patterns generated in real time by a program called CrystalMaker. (See App. C for the details of this comparison.)

Chapter 4

Simulations using Spinach

The quantum dynamics simulations produced for this project are created using a MATLAB package called Spinach [24] which is used to generate ‘fast’ quantum spin dynamics simulations. Many-body quantum spin dynamics problems may be computationally intractable because of the 2^N dimensional Hilbert space which becomes unmanageable quickly with increasing number of particles. The utility of this simulation package is its ability to use restricted basis sets which reduce the overall processing time required to evolve the system through the desired number of time steps. Although the maximum number of active (spin-1/2) particles is 12 with our current computation ability, periodic boundary conditions can be imposed on the system which, to some degree, are able to capture the bulk behavior of the material. The agreement between the experimental line shapes obtained in this project (see Ch. 5 for details) and primary features in line shapes generated by Spinach spin dynamics simulations (described below) indicate that the finite particle number limitation can be addressed computationally with several techniques. These techniques include disorder averaging and an adaptive lattice generation technique which will be described in detail in Sec. 4.2.

4.1 Rotations and orientations

Rotations are important in many different branches of mathematics and physics. In this project rotations are important because they provide a way to build clusters of atoms in a simulated (finite) silicon crystal lattice and rotate them into any one of the four orientations of interest

mentioned in Sec. 3.2.

There are many different ways to specify rotations in 3-dimensional space. There are active and passive rotations, in which either the coordinate system or the coordinates specifying an object in space is rotated. There are also several (nearly) equivalent conventions for rotation. The convention used in this experiment is presented below.

There are two defining features of a rotation matrix:

- (1) The determinant is unity

$$\det(\mathbf{R}) = 1$$

- (2) The inverse of the matrix is also its transpose

$$\mathbf{R}^T = \mathbf{R}^{-1}$$

Since the determinant is unity, no scaling occurs when the rotations are applied to an arbitrary vector. Although the unitarity principle can be extended to higher dimensions, the physically relevant vector space is \mathbb{R}^3 , so it is the only one explicitly considered here. Since there are three spatial dimensions, three linearly independent basis vectors are needed to span the \mathbb{R}^3 vector space.

In an \mathbb{R}^3 vector space with three orthonormal basis vectors given as $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ and cyclic permutation given by the right hand rule, a matrix \mathbf{P} can be constructed which represents a rotation from its starting position back onto itself (i.e., the 3x3 identity matrix \mathbb{I}_3).

$$\mathbf{P} = \mathbb{I}_3 = \{\hat{e}_1, \hat{e}_2, \hat{e}_3\} \doteq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \hat{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \hat{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \hat{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.1)$$

A right-handed rotation about the x-axis, y-axis, or z-axis is encoded in the matrices given in Eqs. 2.62 and 2.63. These rotations are, by convention, the same as \hat{e}_1, \hat{e}_2 , or \hat{e}_3 , respectively.

Such a rotation will take the original components in as a vector multiplied on the right and produce a resultant vector with the new rotated components as three column-wise entries.

For example, to rotate the vector

$$\vec{\mathbf{v}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \hat{e}_1 \quad (4.2)$$

about the positive z-axis (ccw as viewed from above, according to the right-hand rule) by an angular displacement of $\phi = 90^\circ$ use

$$\mathbf{R}_{+z,90^\circ} \vec{\mathbf{v}} = \begin{bmatrix} \cos(90^\circ) & -\sin(90^\circ) & 0 \\ \sin(90^\circ) & \cos(90^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \hat{e}_2. \quad (4.3)$$

This is the expected result as the rotation of the x-axis unit normal vector by 90° about the z-axis is expected to produce the y-axis unit normal vector.

To obtain a general rotation about a fixed origin point, these matrices can be applied in sequence by multiplying a coordinate vector from the right, starting with the first rotation.

To continue the example above, if we wanted to rotate the x-axis unit normal vector first by 90° about the +z-axis, and then additionally we want to rotate this vector by 90° about the -x-axis (left hand rotation, or clockwise as viewed from positive to negative direction along the x-axis).

The expression representing this rotation is given by

$$\begin{aligned} \mathbf{R}_{-x,90^\circ} \mathbf{R}_{+z,90^\circ} \vec{\mathbf{v}} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-90^\circ) & -\sin(-90^\circ) \\ 0 & \sin(-90^\circ) & \cos(-90^\circ) \end{bmatrix} \begin{bmatrix} \cos(+90^\circ) & -\sin(+90^\circ) & 0 \\ \sin(+90^\circ) & \cos(+90^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} = -\hat{e}_3, \end{aligned} \quad (4.4)$$

which is the expected result.

These rotations are passive rotations which transform the components of an arbitrary vector in space, while preserving the orientation of the coordinate system. The rotations are taken to be

right handed rotations, where the positive rotation is defined as the direction of a right handed rotation about a specified axis. This will continue to be the standard basis for rotations in this chapter.

An appropriate rotation scheme can be specified by the three angles of rotation α , β , and γ , which represent rotations about the \hat{z} -, \hat{y} -, and then \hat{z} -axis again. The general rotation matrix is given in Eq. 2.63 in this convention. To apply the rotation specified in this section to a vector use

$$\vec{x}' = R\vec{x}, \quad (4.5)$$

and since a vector basis rotation can be thought of as three separate vector rotations, the rotation of a basis set of three vectors given by $\mathbb{B} = [\vec{x}_1, \vec{x}_2, \vec{x}_3]$ is accomplished by

$$\mathbb{B}' = R\mathbb{B}R^T. \quad (4.6)$$

4.2 Lattice construction

Silicon crystals exhibit the diamond cubic crystal structure, which follows the face-centered cubic Bravais lattice with additional atoms located at four of the midpoints between the center of the unit cell and the corners. The primitive unit cell has translation vectors which connect one of the corners of the unit cell to each of the three conventional cell face centers. There are two atoms in the primitive unit cell, one at the origin (i.e., any corner of the conventional unit cell), and one at one quarter of the length of the body diagonal of the conventional unit cell (measured from the origin). Recall that the lattice constant for single-crystal silicon is $a_0 = 5.43096 \text{ \AA}$. The primitive lattice translation vectors are then given explicitly as

$$\vec{a}_1 = \frac{a_0}{2} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \vec{a}_2 = \frac{a_0}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \vec{a}_3 = \frac{a_0}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad (4.7)$$

and the coordinates of the two atoms in each primitive unit cell are given as

$$\vec{r}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \vec{r}_2 = \frac{a_0}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \quad (4.8)$$

The quantum spin dynamics simulations created in Spinach for this project use an adaptive lattice construction technique described below (see the MATLAB code in Sec. E.2). The original algorithm for assigning coordinates to active and inactive atomic sites in the lattice assumed a fixed $4 \times 4 \times 4$ primitive cell cluster size. In practice, there were many configurations with only a single spin in the 128 spin cluster (2 atomic sites per unit cell, 64 primitive cells). This is computationally inefficient; thus, the adaptive cluster technique is designed to produce clusters which are expressible as integer multiples of the primitive lattice translation vectors and also include as many active atoms as possible without exceeding the maximum number of allowed spins n_{max} .

The method is called adaptive because the number of unit cells along a certain primitive translation vector axis is increased in each iteration, and the cluster of the specified size is populated with uniformly random isotopes at natural isotopic abundance for silicon. In each iteration the number of active atoms is counted. If the number of active atoms exceeds the maximum number of allowed active atoms n_{max} , then the previously built lattice is used instead. This will produce clusters which are much larger and also include many more active atoms in the configurations. Larger cluster sizes with more spins will reduce the number of simulation trials necessary to capture higher orders of spin-spin interactions which more closely model the experiment. However, this will come at the cost of an increase in Hilbert-space dimension for the system which increases the necessary number of computations.

The adaptive cluster builder algorithm proceeds as follows:

- (1) Define the primitive lattice translation vectors as above. Note that in the rotating frame there is rotational freedom in definitions for the x and y axes, but the z-axis must remain vertical and in the \hat{e}_3 position.

- (2) A maximum number of active nuclei n_{max} is specified, as well as the minimum and maximum number of primitive cells to include along each translation axis (specified by one of the primitive lattice translation vectors) in the form $[a_{1,min}, a_{2,min}, a_{3,min}]$ and $[a_{1,max}, a_{2,max}, a_{3,max}]$.
- (3) A one-dimensional array is then created with length equal to the total number of spins in the system specified by the minimum number of cells along each translation axis and the number of atomic occupation sites in each primitive cell (for silicon, this is $n_{occupation} = 2$) as $n_{sites} = a_{1,min} \times a_{2,min} \times a_{3,min} \times n_{occupation} = 2(a_{1,min} \times a_{2,min} \times a_{3,min})$. This array is then populated by random numbers uniformly distributed between zero and one. Since the probability of a site being occupied by an active nucleus is 4.6832%, a logical test is performed at each site to find whether the randomly assigned value is less than 0.046832. If this logical test evaluates to true, then the corresponding array value is changed to '1' to indicate that the site is filled by a magnetically active spin-1/2 ^{29}Si atom. Otherwise, the array entry is changed to '0' to indicate that the site is filled with a passive spin-0 isotope of the silicon atom.
- (4) A **while** loop is then run with the termination condition that either the number of sites which are occupied by active spin-1/2 nuclei is greater than the specified n_{max} or one of the integers specifying the number of translations along a primitive lattice translation vector is greater than the allowed maximum value for that particular axis.
 - (a) Each time the **while** loop is run, an empty array is created for the coordinates of each atomic site in the lattice and also one for the coordinates of the atomic sites which are occupied by an active spin-1/2 ^{29}Si atom. Also the number of total atomic occupation sites n_{sites} and the number of sites occupied by active nuclei n_{active} are both initialized to zero each time the while loop is run.
 - (b) A set of four nested **for** loops is then run which is the true generator of the lattice, in the sense that this is where the full coordinates array and active site coordinates array are both populated. Three of the **for** loops are run over the indices specified

initially by the integers $[a_{1,min}, a_{2,min}, a_{3,min}]$, which are then incremented according to a scheme described below. The fourth **for** loop is run over the number of atomic occupation sites in each primitive cell (again, for silicon this is $n_{occupation} = 2$).

- (c) During each pass of the nested **for** loops, a coordinate array is created and assigned the three values corresponding to the (x, y, z) Cartesian coordinates of the vector given by $\vec{r} = a_1\vec{a}_1 + a_2\vec{a}_2 + a_3\vec{a}_3 + \vec{r}_i$, where \vec{r}_i is the vector which corresponds to one of the possible occupation sites of the atoms in the primitive lattice and the \vec{a}_i vectors correspond to each of the primitive lattice translation vectors given above in Eq. 4.7. The \vec{r}_i vectors are given explicitly for silicon in Eq. 4.8.
- (d) These coordinate vectors which are created during each pass of the nested **for** loops are stored in order in the (now non-empty) full coordinate arrays which are initialized to zero at the start of the **while** loop pass. This array stores the coordinates of all possible atomic sites in the lattice, whether or not the sites are occupied by an active spin-1/2 nucleus. A running index of the site number (corresponding to the number of passes made by the nested **for** loops to that point) is able to correlate the array of '0's and '1's to the coordinate created by the current pass of the nested **for** loops. A logical test is performed to discern whether or not the site corresponding to the current coordinate should be labeled as being filled with an active spin-1/2 ^{29}Si atom or not. If the test shows that the site is filled with an active atom, then the counter variable for active sites is incremented and the coordinates are added to the array with the coordinates of only active atoms. Both of these arrays of coordinates will be used by the Spinach kernel to perform the spin dynamics simulation described below.
- (e) The array of the coordinates of all atoms in the lattice, the array of the coordinates of only active atoms in the lattice, the total number of atoms in the lattice, and the number of active atoms in the lattice are now defined and stored separately for possible later use. The number $a_i \in \mathbb{Z}$ which indicates the number of primitive cells

to be included in the lattice cluster along the i th primitive translation vector axis is incremented, and the subscript indicating which axis is to be incremented in the next iteration is also incremented. A temporary one-dimensional array is created with number of entries equal to the total number of possible atoms in the lattice given by the above relation $n_{sites} = 2(a_{1,j} \times a_{2,k} \times a_{3,l})$, with $j \in [a_{1,min}, a_{1,max}]$, $k \in [a_{2,min}, a_{2,max}]$, and $l \in [a_{3,min}, a_{3,max}]$.

- (f) This is where the while loop ends and is run again if the tests for total active atom number and maximum primitive cell number along any dimension evaluate to true. Once the test evaluates to false, the coordinates of all atom sites and the active atom sites are still recorded in the separate variables mentioned above.

Although the adaptive cluster algorithm is fully functional, the original simulation code used a fixed cluster size of $4 \times 4 \times 4$ which (with 2 atomic sites per lattice) becomes 128 total atoms in the lattice. For the simulations generated in this project, all lattices are fixed at this 128-site size. This is done by setting the minimum and maximum cell dimension vectors to the same $[4,4,4]$ size.

The active and inactive atomic sites in the silicon lattice model and stick representations of the J-coupling tensors between nearest neighbors are shown in Fig. 4.1.

4.2.1 Disorder Averaging

The Spinach spin dynamics simulation package is able to efficiently perform the various computations needed to simulate a spin dynamics experiment (in the context of NMR spectroscopy for this project), but it is still unable to exactly model the many body physics of a system with a large number of interacting particles. The real physical system being modeled contains a macroscopic ($\approx 10^{23}$) number of interacting particles in the experimental sample described in Ch. 3. The maximum number of interacting particles which can be modeled in the Spinach spin dynamics simulations is approximately 20, depending on the available processing ability. For desktop computers, the maximum number of particles is about 10. This is the value for n_{max} used in this project.

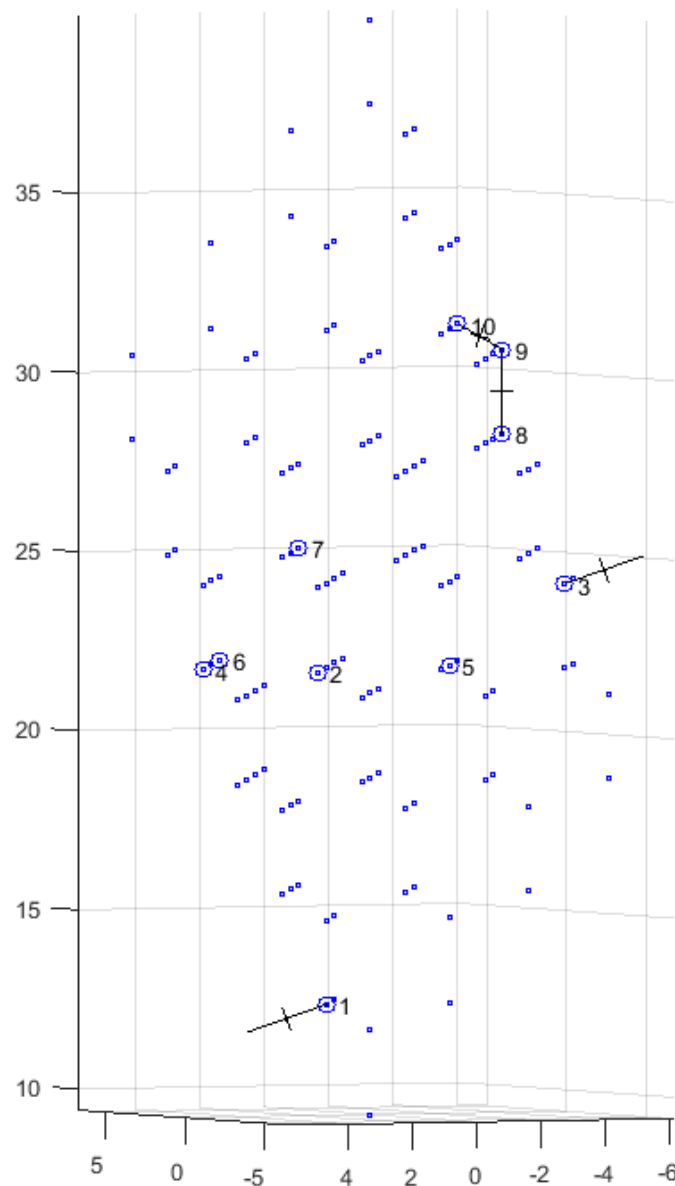


Figure 4.1: Three-dimensional plot of simulated cluster. This plot shows an example of a finite silicon lattice cluster generated for the spin dynamics simulations in this project. Randomly assigned active atoms are circled, and nearest neighbors are joined by lines representing the three principle axes of the J-coupling tensor joining the two adjacent atoms, even across the periodic boundary. This is shown between atoms '1' and '3' in this figure.

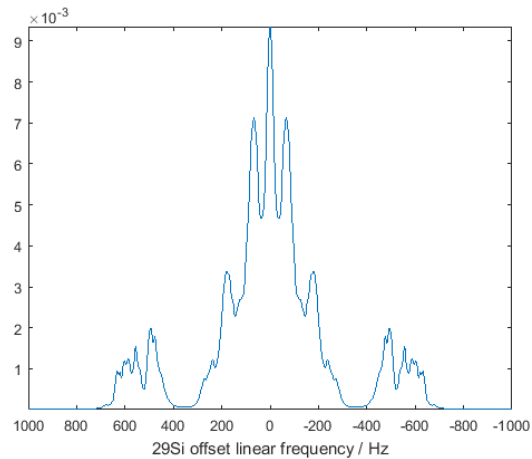
One of the techniques used to circumvent this limitation is disorder averaging which involves the generation of a large ensemble of individual configurations. The line shapes produced by these configurations are then summed to attempt to capture the behavior of the macroscopic spin dynamics without having to compute the evolution of spin systems with many ($n \gg 10$) active spins.

The transformed FIDs produced by small systems (≈ 6 active atoms) typically only produce single peaks since the active atoms are not likely to interact strongly in an extended lattice. The lattices constructed in this project are considered ‘small’ since only 128 total atomic sites are used, which with a 4.6832% occupation of active atoms gives (by the binomial distribution) $\bar{n}_{\text{active}} \approx 5.99$. This is the same number of atomic sites in the lattice used to generate Fig. 5.2, which shows a comparison of the experimentally obtained line shapes to the simulated ones for all four of the orientations of interest in this project. In some configurations, there are strongly interacting active atoms (nearest neighbors or small order of atomic separation) which produce the pronounced splittings labeled in Fig. 5.2, as well as other weaker splittings.

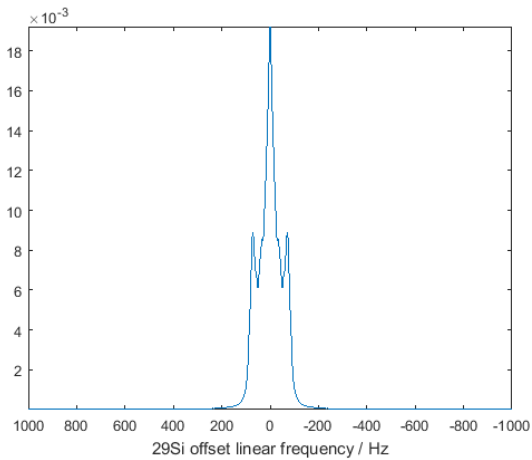
These microscopic configurations do not resemble the experimental line shape, and are also quite different from one another. Figure 4.2 shows the line shapes produced by three different clusters of the same size with randomly assigned atomic occupations to demonstrate some possible deviations from the experimentally obtained line shapes. When a large ensemble of these configurations are used to run identical simulated experiments, then the simulated FIDs can be summed together to try to capture the macroscopic behavior of the dynamical system. Since the spin-spin interactions are dependent on separation distance between active nuclei, it is reasonable to assume that the macroscopic system can be modeled accurately by summing a representative number of smaller order interactions.

4.3 Simulation properties and organization

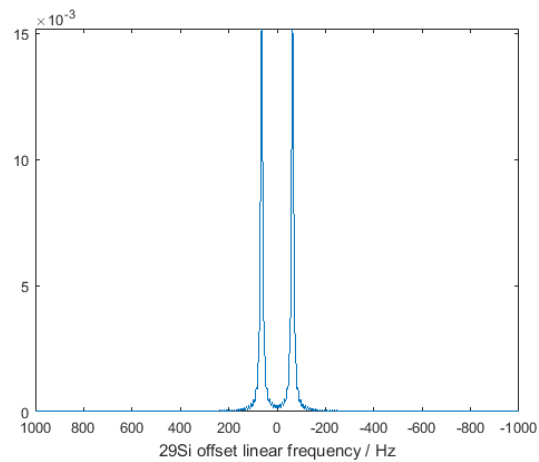
The simulations are run using the Spinach kernel as in the code attached in App. Sec. E.1. The simulation is organized so that the following control parameters are defined initially, including



a) 10 active atoms in the configuration



b) 7 active atoms in the configuration



c) 2 active atoms in the configuration

Figure 4.2: FIDs generated by single configurations. Line shapes from individual configurations from the disorder average ensemble. These are summed together and weighted appropriately to model the spin dynamics of the macroscopic sample.

the following:

- Random number generation and batch serialization control parameters
- Disorder average ensemble member number
- Maximum number of active atoms in any configuration
- Minimum and maximum number of cells along any translation vector
- Dipole-dipole interaction cutoff length (15 Å)
- J-coupling interaction cutoff length (3 Å)
- J and ΔJ values in Hz
- Strength of the external B field in Teslas
- Active nucleus isotope label (for Spinach function calls)
- Lattice constant a_0
- Isotopic abundance percentage for active nucleus
- Apodization to be applied in Fourier transform of FID generated by Spinach
- Primitive cell translation vectors
- Relative coordinates of atoms in each unit cell
- Primitive cell volume

The heart of the simulation is performed in a **for** loop over the specified number of throws.

The main structure of the **for** loop is as follows:

- (1) Initialize the random seed (for reproducibility) according to specified control parameters
- (2) Create the lattice configuration objects (see Sec. E.2 for code), including:

- Coordinates of all atomic sites
 - Coordinates of active atomic sites
 - Number of active atomic sites
 - Number of total atomic sites
- (3) Create array of J-coupling interaction tensors (3×3 matrices) (see Sec. E.3 for code and Sec. 4.3.1 for a description)
 - (4) Specify filename for log file (generated for each simulated experiment)
 - (5) Set the Zeeman interaction strength to zero
 - (6) Set the temperature to 298K
 - (7) Specify the periodic boundary conditions by giving the primitive translation vectors
 - (8) Set some of the Spinach specific conditions
 - (9) Create the spin system object for Spinach
 - (10) Report on the system (recorded in the output log file)
 - (11) Set the initial state of the system to $(S^+ + S^-)/2$
 - (12) Set the experimental sequence parameters
 - State of the coil
 - Offset frequency of coil
 - Frequency spectrum to sweep (in Hz)
 - Number of points to sample
 - Zero filling
 - Plotting specifications

- (13) Run the experiment for the four different orientations of interest in parallel using the ‘powder.m’ experiment function (included in the Spinach library)
- (14) Save the FIDs
- (15) Optionally plot the cluster using the `cluster_plotter_v1.m` function (available upon request)
- (16) Save experimental parameters for each run
- (17) Initiate `fidSum_v5.m` to prepare and plot the FIDs generated in the main body of the code (see Sec. E.4 for code and Sec. 4.3.2 for a description)

4.3.1 Including the J-coupling interaction

The J-coupling interaction is included in the simulation separately from the dipole-dipole interaction. (See the function code in App. Sec. E.3.) The J and ΔJ values are specified in Hz, which are then used to populate the J-coupling interaction tensor (3×3 diagonal matrix in the principal axis basis). Since the approximation is being made that only nearest neighbors have any significant J-coupling interaction, the proximity of all active atoms in the configuration must be tested against the nearest neighbor distance. A connectivity matrix can be built this way, with zeros to indicate no nearest neighbor interaction, and ones to indicate that two active atoms are also nearest neighbors. Since periodic boundary conditions are used to more accurately model the quantum spin dynamics, all active atoms are translated into corresponding positions in each of the 26 surrounding cell clusters of equal primitive cell dimension. The same test for nearest neighbors is performed between all of these translated atoms and the atoms in the original cluster.

Once the connectivity matrix has been populated, the J-coupling interaction tensor in the principal axis basis between each pair of interacting active atoms can be rotated to match the orientation of the bond between them. Each of these tensors (3×3 matrices) is stored in a $n_{active} \times n_{active}$ array which is used by Spinach in the spin dynamics simulations.

4.3.2 Processing the FIDs and plotting

Once the FID data has been obtained by the main body of the simulation code, all of the FIDs are summed together pointwise. (See the code in Sec. E.4.) The FIDs corresponding to each orientation of interest are then apodized, Fourier transformed, and normalized by dividing every point in the summed FIDs by the total number of active atoms present in all contributing configuration. These modified FIDs are then saved and plots are made in the transform domain of the simulated experimental line shape.

Chapter 5

Experiment and Analysis

A standard NMR spectroscopy experiment known as the pulse-acquire experiment is conducted by applying a transverse RF signal, as introduced in Ch. 2, and then the same coils are used to measure the transverse magnetization excited in the sample. This detected magnetization is also known as the free induction decay (or FID), which can be transformed into the frequency domain by use of the Fourier transform to obtain the line shape of the experimental system.

This line shape provides information on the interactions between atoms in the sample. There are several standard and equivalent units used to represent the frequency domain. The frequency domain units used in this project are Hz, measured relative to a frequency center at the highest point of the [100] orientation central peak. All peak heights are normalized to the same area as that under the [100] orientation peak.

5.1 The pulse-acquire experiment

The pulse-acquire experiment is the simplest of the NMR spectroscopy experiments which consists of applying a single RF pulse to a sample in thermal equilibrium with the bath, followed by a period of data acquisition. There is an extremely brief ($\approx 1 \mu\text{s}$) delay between the pulse and acquisition. During the acquisition period, the experimental data are collected according to the specified sampling rate (also known as ‘sweep width’), desired number of points in the FID, and the offset frequency. The transverse magnetization induced by the RF signal is recorded at each of the specified time steps in the experiment, and is saved for later manipulation and refinement.

If the spin of an active atom is visualized as a vector in three-dimensional space, then the orientation of this vector in thermal equilibrium will be the same as that of the externally applied magnetic field. In this project the \hat{z} -direction is the assumed orientation for the external field. When the RF pulse is applied, it has the effect of rotating the spin vector about the axis of rotation. In this project, all rotations are assumed to be about the \hat{x} -direction. The angle of rotation of the spin vector is dependent on the strength and duration of the applied RF signal. A stronger pulse or one that is applied for a longer duration will produce more rotation in the spin vector than a weaker or shorter pulse. Two of the most common types of pulses used for the simple pulse-acquire experiments are pulses which correspond to 90° and 180° rotations of the spin vector about the axis of rotation.

5.2 T_1 measurement

The T_1 thermalization time, as introduced in Sec. 2.4.2, is related to the strength of the interactions of the spins with the thermal bath. The method for determining the T_1 relaxation time in this project is to perform a series of simple pulse-acquire experiments. In each experiment, the sample is prepared so that the spins are not in thermal equilibrium but are rather in a disorganized state with no overall magnetization in the \hat{z} -direction. This is done by applying a series of 90° pulses which rotates any magnetization in the \hat{z} -direction into the transverse plane. The time required for the magnetization in the \hat{z} -direction to return to a significant fraction ($1/e$) of its maximum value is the T_1 thermalization time.

After the magnetization in the \hat{z} -direction has been rotated into the transverse plane, the sample is allowed to return (slowly in this project) to equilibrium. When enough time has elapsed to allow a significant portion of the magnetization to equilibrate, a second pulse-acquire experiment is performed with another 90° pulse to rotate any accumulated magnetization in the \hat{z} -direction into the transverse plane where it will precess and produce an oscillating magnetic field which can be detected by the RF coils and then recorded as a FID measurement.

The first pulse-acquire experiment zeros the magnetization in the \hat{z} -direction by rotating it

into the transverse plane, and the second pulse-acquire experiment measures the induced magnetization in the \hat{z} -direction accumulated during that period. This time interval can be denoted by Δt_i , where i is in reference to the index of the experiment. The T_1 relaxation time can then be determined by fitting the strength of the transverse magnetization recorded after thermalization time Δt_i (for experiment i) for some number of experiments where Δt_i is varied. This method is used to determine the T_1 relaxation in single-crystal silicon in this project, and the results are given in Fig. 5.1. The value for the T_1 relaxation for single-crystal silicon averaged over four important orientations (with little variance between orientations) is found to be $T_1 = 5.6$ hrs.

5.3 Line shape of ^{29}Si in single-crystal silicon

The primary goal of this project is to understand the features exhibited by the NMR line shape of ^{29}Si in single-crystal silicon and the orientation dependence of these features. In the experimentally obtained line shapes, there will be a single large central peak which is minimally distorted by susceptibility mismatch. The magnetically active atoms are all at magnetically equivalent sites in the silicon lattice, and therefore the line shape will have only a single central peak.

There are also other smaller features which are expected to be present in the NMR line shape that depend on the orientation of the silicon lattice with respect to the externally applied magnetic field. These smaller features are the result of the dipole-dipole interaction which occurs between two active atoms. A discussion of these orientation dependent splittings will be given below. The third type of interaction discussed in Ch. 2 is the J-coupling interaction which is a weaker interaction compared with the other two. The effect of the J-coupling is to reduce the effective splitting frequency of the nearest neighbor dipole-dipole interactions. This effect will also be described in some detail below.

When the experiments are performed in the NMR spectrometer and FIDs are recorded, processed, and transformed into the frequency domain for each of the given orientations, the line shapes shown in Fig. 5.2 are produced and compared to the simulated line shape modeled in the Spinach spin dynamics simulations. The angle markings correspond to different nearest neighbor

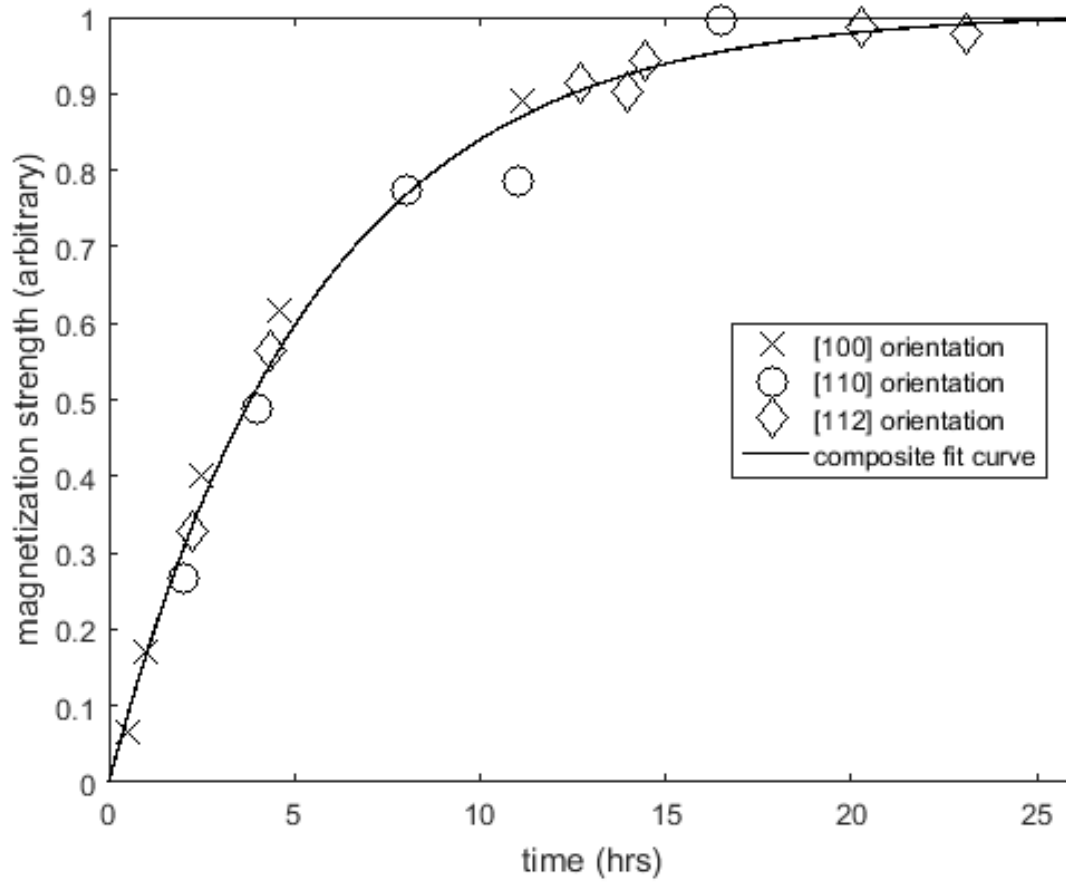


Figure 5.1: T_1 measurements. T_1 Measurements and a fit of all data points for the three labeled orientations are shown. The T_1 value obtained from this fit is 5.5 hours.

pairs in the different orientations (see Sec. 5.3.1 below for details).

5.3.1 Derivation of splitting size

The splittings produced by the interaction of two magnetic dipoles i and j in the presence of a large external magnetic field can be calculated using the formula

$$\Delta f = \begin{cases} \left| \left(\Delta J + \frac{3}{2\pi} \right) \frac{-\mu_0 \gamma^2 \hbar}{4\pi r_{ij}^3} \left(\frac{3 \cos^2 \theta_{ij} - 1}{2} \right) \right|, & \text{nearest neighbor pair} \\ \left(\frac{3}{2\pi} \right) \frac{-\mu_0 \gamma^2 \hbar}{4\pi r_{ij}^3} \left(\frac{3 \cos^2 \theta_{ij} - 1}{2} \right), & \text{otherwise,} \end{cases} \quad (5.1)$$

with the J-coupling only applied to the nearest neighbor pairs as described in chapter 2 since

- (1) the interaction falls off with distance, and
- (2) the strength of the nearest neighbor J-coupling interaction is approximately $1/10^{th}$ as strong as the dipole-dipole interaction (see Sec. 5.3.2 for details).

These splittings are also referred to as doublets since they produce two equally spaced peaks about the center frequency. Positive and negative splittings (doublets) will both produce the same splittings in the line shape, so they are considered to be equivalent. The bond lengths r_{ij} , angles between the bonds and externally applied field θ_{ij} , and the frequency splittings for the [100] and [111] orientations are given in Table 5.1 and for the [112] and [110] orientations in Table 5.2. Each of these splittings are predicted to appear in the experimental line shape, although the strength of the interaction should be greater for nearest neighbors than for those further away since the dipole-dipole interaction falls off as the inverse cube of the separation distance. These splittings are shown in Fig. 5.3.

5.3.2 Effect of the J-coupling interaction

The effect of the J-coupling interaction can be seen clearly when the interaction is removed in the spin dynamics model and compared to the experimental spectrum. This is shown in Fig. 5.4. From this comparison, the value of ΔJ defined in Ch. 2 is found to be

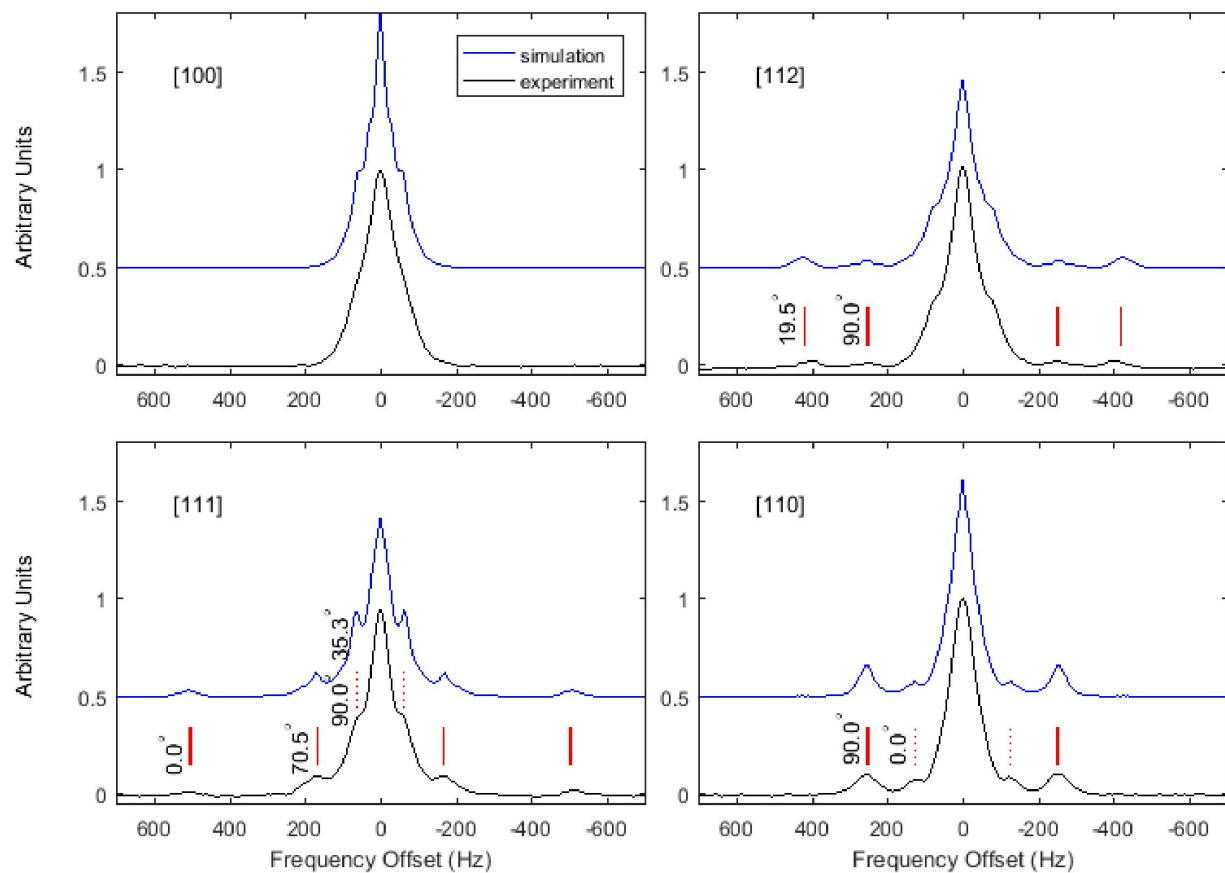


Figure 5.2: Experimentally obtained line shapes in all four orientations with simulated line shapes for comparison. Experimentally obtained line shapes in all four Experimental and simulated line shapes (offset above) for the four orientations of interest in this project. Pairs of peaks due to nearest neighbors (solid red lines) and next nearest neighbors (dotted red lines) are labeled with the angle of separation between a vector connecting the two active atoms involved in the interaction and the direction parallel to the externally applied magnetic field.

[100] Orientation				[111] Orientation			
Bond Type	Number	Angle (°)	Δf (Hz)	Bond Type	Number	Angle (°)	Δf (Hz)
nn	4	54.74	0.1	nn	1	0.00	-1005.2
nnn	4	90.00	125.8	nn	3	70.53	335.1
nnn	8	45.00	-62.9	nnn	12	90.00	125.8
nnnn	8	72.45	56.7	nnnn	3	29.50	-99.3
nnnn	4	25.24	-113.4	nnnn	3	79.58	70.3
edge	2	0.00	-88.9	nnnn	6	58.16	12.9
edge	4	90.00	44.5	edge	6	54.74	0.0

Table 5.1: Bond lengths, angles, and splitting frequencies in the [100] and [111] orientations. The orientations are relative to the externally applied magnetic field, as usual. The bond types are nearest neighbor (nn), next nearest neighbor (nnn), next next nearest neighbor (nnnn), and next next next nearest neighbors (edge, since these bonds are also the length of one edge of the lattice). The corresponding distances are $\text{edge}=5.43096 \text{ \AA}$, $\text{nn}=\text{edge}*\sqrt{3}/4 \approx 2.35 \text{ \AA}$, $\text{nnn}=\text{edge}*\sqrt{2}/2 \approx 3.84 \text{ \AA}$, $\text{nnnn}=\text{edge}*\sqrt{11}/4 \approx 4.50 \text{ \AA}$. The number column gives the number of atoms of that bond type which are at the same angle with respect to the externally applied magnetic field. The angles column gives the angle formed between the bond and the externally applied magnetic field and is given in degrees. The frequency splittings Δf are given in Hz and represent the frequency difference between two of the side peaks found in the transform of an experimentally obtained FID. Note that negative splitting values appear the same as positive splittings in the line shape by symmetry.

[110] Orientation				[112] Orientation			
Bond Type	Number	Angle (°)	Δf (Hz)	Bond Type	Number	Angle (°)	Δf (Hz)
nn	4	90.00	502.6	nn	1	19.47	-837.7
nnn	2	0.00	-251.5	nn	1	90.00	502.6
nnn	8	60.00	31.4	nn	2	61.87	167.4
nnn	2	90.00	125.8	nnn	4	30.00	-157.2
nnnn	2	90.00	78.0	nnn	2	54.74	0.0
nnnn	6	64.76	35.5	nnn	2	90.00	125.8
nnnn	4	31.48	-92.2	nnn	4	73.22	94.3
edge	2	90.00	44.5	nnnn	4	42.39	-49.6
edge	4	45.00	-22.2	nnnn	3	60.50	21.3
				nnnn	2	90.00	78.0
				nnnn	2	75.75	63.8
				nnnn	1	10.02	-148.9
				edge	2	35.26	-44.5
				edge	4	65.91	22.2

Table 5.2: Bond lengths, angles, and splitting frequencies in the [110] and [112] orientations relative to the externally applied magnetic field, as in Table 5.1. The convention used to denote the bond length by type as in Table 5.1 is also used here.

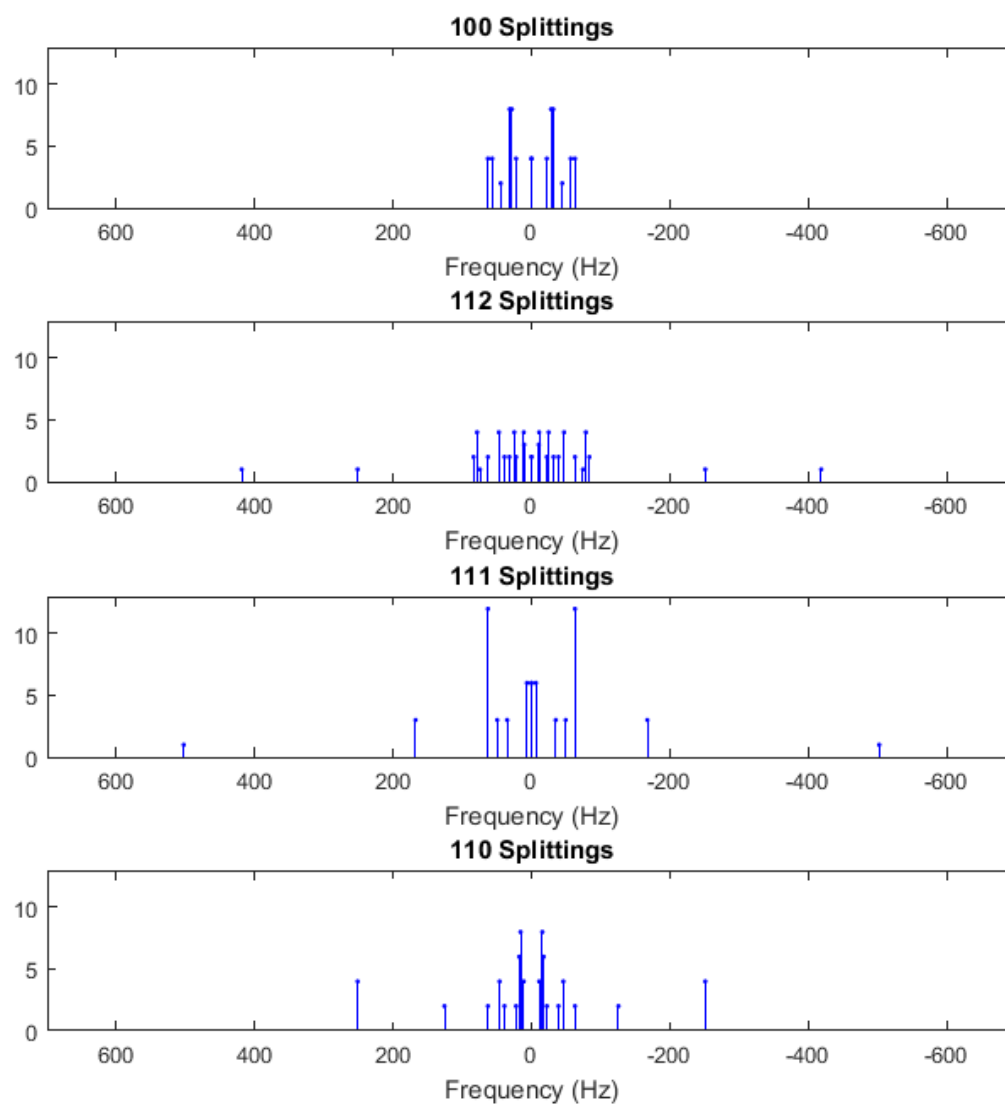


Figure 5.3: Frequency splitting stem plots for all four orientations. The frequency of each splitting is marked with the number of atoms which share that same splitting on the vertical axis. Note that the splittings are symmetrically distributed about the origin.

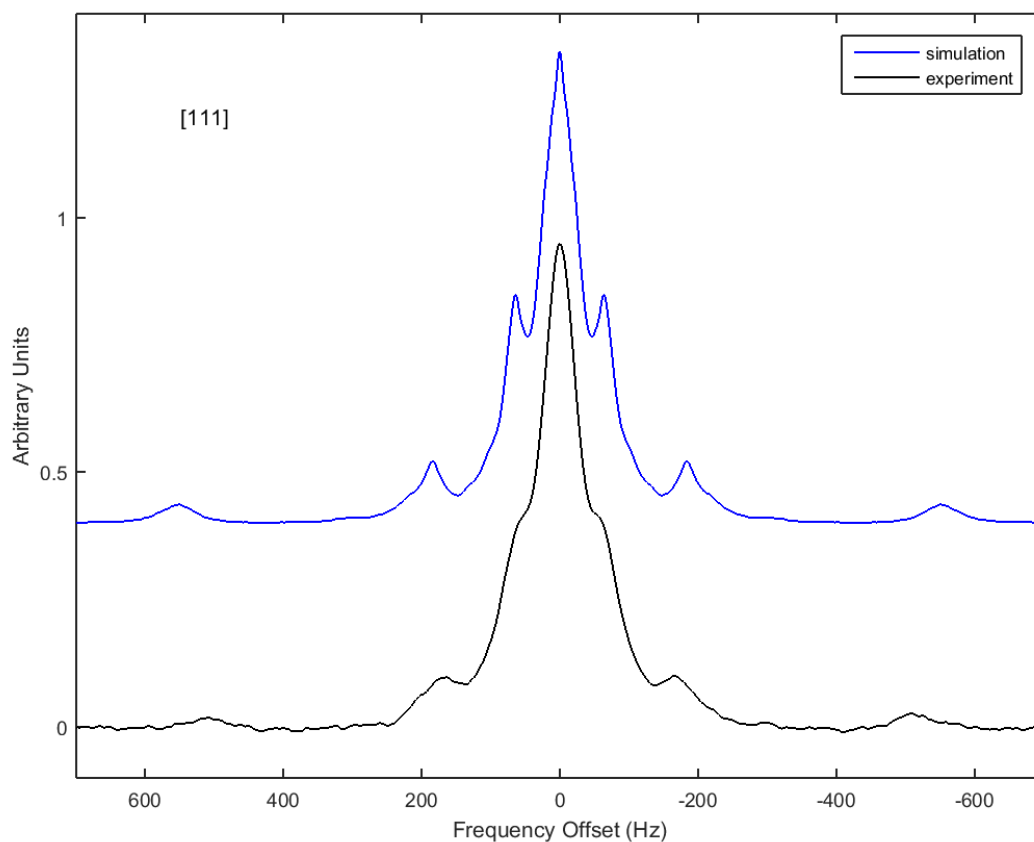


Figure 5.4: J-coupling effect. The effect of ignoring the J-coupling interaction in the Spinach spin dynamics simulation is to increase the splittings of nearest-neighbor doublets. Here the widest doublet in the simulated line shape is produced by the nearest neighbor in-line with the externally applied magnetic field. The simulated splitting, which ignores the J-coupling interaction, is nearly 1100 Hz wide where the experimental line shape exhibits a splitting of only 1000 Hz. This disparity is used to fit the value of ΔJ defined in Ch. 2.

$$\boxed{\Delta J = 90 \pm 20 \text{ Hz}} \quad (5.2)$$

The J-coupling interaction is caused by screening effects from the electrons surrounding the silicon atoms which mediate the magnetic interaction between the magnetically active nuclei. This has the effect of reducing the splitting frequency, but otherwise has no significant effects on the line shape.

5.4 Comparison of experimental results to model

The experimentally obtained line shapes are shown in Fig. 5.2 for each of the four orientations considered in this project. As mentioned before, there are markings which indicated which of the arrangements of bonds is producing the wide doublets.

Above each of these is a plot of the line shape obtained by Spinach simulation and disorder averaging over an ensemble with 100 members. The line shapes produced by this method exhibit the primary features noted above in the experimentally obtained line shapes. The large central peaks representing the isotropic chemical shift and the side peaks corresponding to the calculated frequency splittings are represented, although the simulated peaks are more jagged compared to the experimentally obtained ones. This relative roughness is likely caused by an insufficient number of elements in the disorder ensemble, or number of spins. The convergence of the line shapes produced by the model and simulation to those obtained experimentally is discussed in further detail in the next section.

5.4.1 Model Convergence

The method described in Sec. 4.2 has three parameters which are used to characterize it in the context of the Spinach simulations. These properties are the number of elements in the disorder ensemble, the number of cells in each cluster, and a parameter used by Spinach to specify the maximum number of product spin operators in the basis. This last parameter indicates that if, for example, the maximum level is specified to be five, then the basis will consist of all single

spin operators, products of two spins, etc., *and* products of all five spins. This exclusion of higher order correlations is offset by corrections during propagation which only allow the system to be propagated into the specified basis.

Each of these three parameters can be adjusted in the simulation to produce corresponding line shapes. These can then be compared against each other and ultimately against the experimentally obtained line shapes to determine whether the model is converging to the desired line shape. The agreement between the simulated and experimentally obtained line shapes shown in Fig. 5.2 suggests that if proper corrective measures are taken to address the finite cluster size problem, then the model will in fact converge to the experimentally obtained line shape within some allowed tolerance related to the assumptions made about the relative strengths of the interactions discussed in Ch. 2.

The convergence of the model with respect to each of these three parameters is shown graphically in Figs. 5.5, 5.7, and 5.6. The default values used are 64 cells in the cluster, 100 members in the disorder average ensemble, and up to 11 simultaneously interacting active atoms. Each of these parameters is varied one by one and plotted on the same axes and offset from each other vertically in the figures.

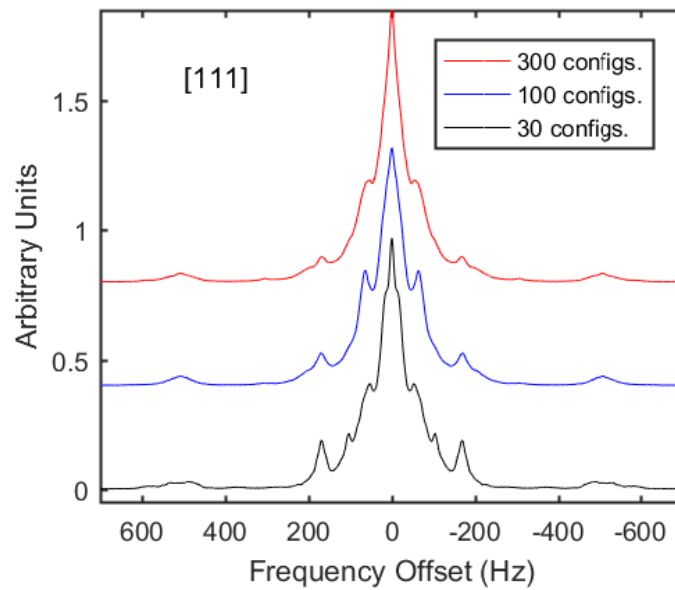


Figure 5.5: Disorder average convergence. Depiction of the convergence of the model with respect to the number of elements in the disorder average ensemble.

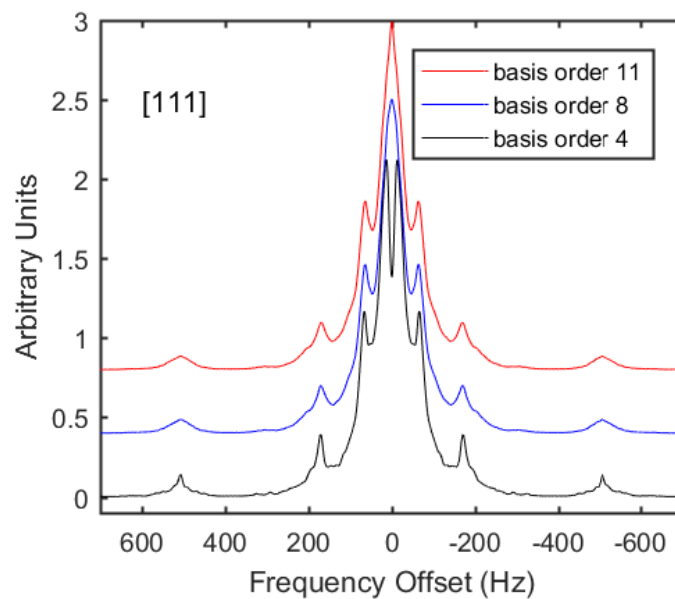


Figure 5.6: Basis level convergence. Depiction of the convergence of the model with respect to the maximum number of product spin operators allowed in the basis.

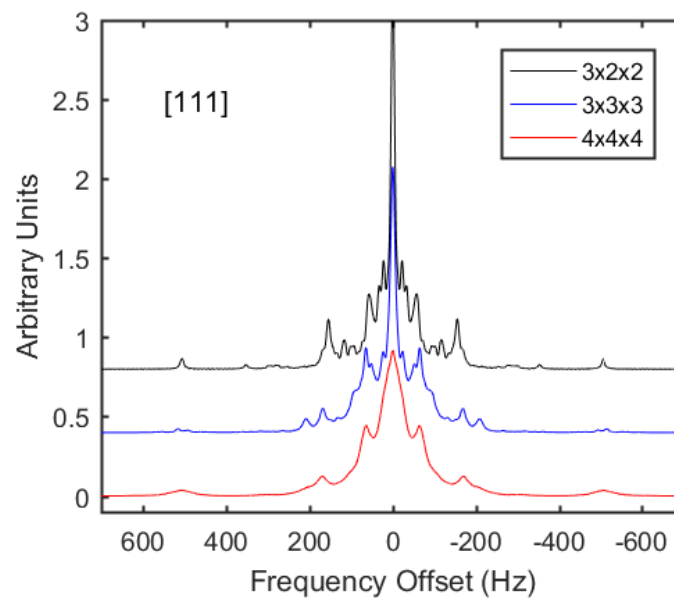


Figure 5.7: Cell size convergence. Depiction of the convergence of the model with respect to the number of cells in the lattice cluster used in the simulations.

Chapter 6

Conclusion and results

The purpose of this project is to understand the dynamical behavior of many interacting spins in a lattice ordered by position and disordered with respect to isotopic occupation. The method of NMR spectroscopy is used to measure the induced magnetization with the pulse-acquire experiment described in Sec. 5.1, and the free induction decay signal is then transformed to obtain the experimental line shape in Fig. 5.2. The system can be modeled using first principles and a disorder average over an ensemble of finite atom clusters to produce simulated line shapes which can be compared to those obtained by experiment.

The effort taken to reduce line broadening in the line shape by the method of susceptibility matching as described in Sec. 3.3 is a novel improvement on other similar experiments which have been performed. This reduction in the line broadening allows for fine features to be revealed in the line shape which have never been accessible before. These fine features correspond to spin-spin interactions which are of primary interest in this project. These features can then be compared to those obtained by simulating these interactions using first principles and a disorder average as explained in Ch. 4.

It has been shown that the model predicts several definite features in the line shape, and that these features are identifiable in the experimentally obtained line shape. This agreement shows that the project goal to understand the interactions of a many-body dynamical spin system has been met within the desired accuracy of the experiment.

The jaggedness of the line shapes produced by the Spinach simulations indicates that the

parameters used in this experiment could be varied further to improve the convergence of the simulated line shape. With the parameters used in this project, however, the important features of the line shape are still identifiable and give good agreement to those found experimentally.

As mentioned in Sec. 3.1, these same experimental techniques can be applied to other materials such as quartz to produce experimental line shapes. These other materials could have attractive features for investigating the physics of spin transport of many-body localization. This project shows that the technique of NMR spectroscopy can be used to understand and possibly manipulate these interactions. It may also be used to understand spin diffusion or spin transport in many body interacting spin systems.

There are several possible avenues for future work on this project. One possibility for extending this thesis is to perform simulations on high performance computing clusters to determine appropriate convergence parameters. Another possible option for additional work on this topic is to track the growth of correlations over longer times. This can be done by experiments involving many-body echoes, similar to those already performed on CaF_2 and other extended systems dominated by magnetic-dipole interactions [25, 26, 27, 28].

Bibliography

- [1] Brooks Christensen and John C. Price. NMR line shape of ^{29}Si in single-crystal silicon, arxiv:1610.03511. Phys. Rev. B, 2016.
- [2] Rahul Nandkishore and David A Huse. Many-body localization and thermalization in quantum statistical mechanics. Annu. Rev. Condens. Matter Phys., 6(1):15–38, 2015.
- [3] James Keeler. Understanding NMR spectroscopy. John Wiley & Sons, 2011.
- [4] D.R. Vij. Handbook of applied solid state spectroscopy. Springer Science & Business Media, 2007.
- [5] Anne S Verhulst, Denis Maryenko, Yoshihisa Yamamoto, and Kohei M Itoh. Double and single peaks in nuclear magnetic resonance spectra of natural and ^{29}Si -enriched single-crystal silicon. Physical Review B, 68(5):054105, 2003.
- [6] Anne S Verhulst. Optical pumping experiments to increase the polarization in nuclear-spin based quantum computers. PhD thesis, stanford university, 2004.
- [7] R. R. Gupta and M. D. Lechner. Nuclear magnetic resonance (NMR) data, volume 35. Springer, Berlin, 2006.
- [8] Frank Uhlig and Heinrich Chr Marsmann. ^{29}Si nmr some practical aspects. Gelest Inc., <http://web.chem.ucsb.edu/~nmr/docs/29Si.pdf>, 2008.
- [9] Hiroshi Hayashi, Kohei M Itoh, and Leonid S Vlasenko. Nuclear magnetic resonance linewidth and spin diffusion in ^{29}Si isotopically controlled silicon. Physical Review B, 78(15):153201, 2008.
- [10] FS Dzheparov, DV Lvov, and MA Veretennikov. Spin-spin relaxation in magnetically dilute crystals. Low Temperature Physics, 41(1):9–14, 2015.
- [11] Kohei M Itoh and Hideyuki Watanabe. Isotope engineering of silicon and diamond for quantum computing and sensing applications. MRS Communications, 4(04):143–157, 2014.
- [12] Adrian Auer and Guido Burkard. Long-range photon-mediated gate scheme between nuclear spin qubits in diamond. Physical Review B, 93(3):035402, 2016.
- [13] L Childress, MV Gurudev Dutt, JM Taylor, AS Zibrov, F Jelezko, J Wrachtrup, PR Hemmer, and MD Lukin. Coherent dynamics of coupled electron and nuclear spin qubits in diamond. Science, 314(5797):281–285, 2006.

- [14] T. D. Ladd, J. R. Goldman, F. Yamaguchi, Y. Yamamoto, E. Abe, and K. M. Itoh. All-silicon quantum computer. Phys. Rev. Lett., 89:017901, Jun 2002.
- [15] Kohei M Itoh. An all-silicon linear chain nmr quantum computer. Solid state communications, 133(11):747–752, 2005.
- [16] David H. McIntyre, Corinne A. Manogue, Janet Tate, and Oregon State University. Quantum mechanics: a paradigms approach. Pearson, Boston, 2012.
- [17] Ramamurti Shankar. Principles of Quantum Mechanics. Springer New York, Boston, MA, second edition, 1994.
- [18] David J. Griffiths. Introduction to quantum mechanics. Pearson, New Delhi?, second edition, 2007.
- [19] Daniel V. Schroeder. An introduction to thermal physics. Addison Wesley, San Francisco, CA, 2000.
- [20] Malcolm H. Levitt. Spin Dynamics: Basics of Nuclear Magnetic Resonance. John Wiley & Sons, Chichester, 2 edition, 2008.
- [21] David Jeffrey Griffiths and Reed College. Introduction to electrodynamics, volume 3. prentice Hall Upper Saddle River, NJ, 1999.
- [22] O. Chubar, P. Elleaume, and J. Chavanne. J. Synchrotron Rad., 5:481–484, 1998.
- [23] O. Chubar, P. Elleaume, and J. Chavanne. Radia.
- [24] HJ Hogben, M Krzystyniak, GTP Charnock, PJ Hore, and Ilya Kuprov. Spinach—a software library for simulation of spin dynamics in large spin systems. Journal of Magnetic Resonance, 208(2):179–194, 2011.
- [25] M. Munowitz and A. Pines. Advances in Chemical Physics, volume 66, pages 1–152. John Wiley & Sons, New York, 1987.
- [26] J. Baum, M. Munowitz, A. N. Garroway, and A. Pines. J. Chem. Phys., 83:2015–2025, 1985.
- [27] Gonzalo A. Alvarez, Dieter Suter, and Robin Kaiser. Science, 349:846–848, 2015.
- [28] H. Cho, T. D. Ladd, J. Baugh, D. G. Cory, and C. Ramanathan. Phys. Rev. B, 72:054427, 2005.

Appendix A

UNITY 500 MHz spectrometer operation

A.1 Setting up an experiment

To set up an experiment using the Vnmrj software, at the command line type “e”, then hit the return key to eject the sample holder from the main body of the machine. Set the prepared sample at the top of the room temperature cavity, on the air cushion created by the blowing air. At the command line type “i”, then hit the return key to insert the sample and holder back into the main body of the machine.

To tune the two channels in the broadband probe, connect the BNC cable from each channel in turn to the tuner. Select the appropriate channel and use the digital display on the tuner, as well as the tuning rods at the base of the probe to tune each channel. Reconnect the BNC cables to their proper inputs on the probe.

In the Vnmrj main window press the “Start” tab and then select “Shim” in the submenu on the left side of the lower main window. Select “FID scan” in the top left corner of the menu window and then rescale the FID area to 500.

Starting with the Z_0 setting, and use the largest increment setting (± 100) to increase or decrease the setting value. If the FID scale increases, then continue to adjust the setting value in that direction until it reaches a maximum. Then repeat the process with Z_1 .

After finding the max FID scale value for Z_1 , continue the process with Z_2 , Z_3 , and maybe Z_4 if it seems sensitive to adjustment. This is an iterative process. Ideally, after each pass with a setting the whole process should be restarted from Z_0 , but this would be impractically inefficient.

Generally the lower Z polynomial values should be readjusted once or twice during adjustment and then again at the end. The same sort of process should extend to the X and Y polynomial values after the pertinent Z values are adjusted.

Once the shimming process is complete, a deuterium lock can be established by navigating to “Start”>“Lock” and turning on the lock in the upper left corner of the lower window. Take note of the lock level, it should rise to >20 to be considered locked. First adjust the Z_0 slider bar to the appropriate level for the substance containing the deuterium (further left for light molecular weight, further right for heavier molecular weight). Adjusting the phase bar during this process should help to increase the lock strength. The power and gain may need to be adjusted to improve results; in this project they are set to power: 37 and gain: 30.

Once the lock is established and the coil is shimmed, the parameters of the experiment can be established. Navigate to “Acquire”>“Channels” to assign the experimental channels. For a silicon experiment, use the following settings:

- Nucleus/Freq.: Si29 (99.214 MHz)
- Offset: 0 Hz
- 90 deg. at pwr: 15.00 μs at 58

For a proton experiment (to check shim, etc.), use the following settings:

- Nucleus/Freq.: H1 (499.894 MHz)
- Offset: 0 Hz
- 90 deg. at pwr: 38 μs at 58

Now the details of the pulse sequence can be entered in the “Acquire”>“Acquisition” menu window with the following settings:

- Data: These settings are fairly typical for most experiments
 - * Spectral Width: 50,000 Hz

- * Acquisition time: 1 s
- * Complex Points: 50,000
- Excitation: These settings are for a 24 hour run with 6 shots taken every 4 hours
 - * Relaxation Delay: 1890 s
 - * First pulse: 0 μs
 - * Inter-pulse delay: 6210 s
 - * Observe pulse: 11.70 μs (or 70 degrees)
 - * Calibration: pw90: 15.00 μs at Power 58 dB
- Scans
 - * Requested: 6
 - * Steady State: Unchecked
 - * Block Size: Checked 1
 - * Interleave: Unchecked
- Receiver
 - * Receiver Gain: 50
 - * Auto: Unchecked
 - * Timing (μs): rof2: 24 alfa 6.6

The pulse sequence can be checked before acquisition by selecting “Acquire” > “Pulse Sequence” from the menu in the lower window. Once the experiment has been set up, the green “Acquire” button can be pressed to begin the specified experiment.

Once the acquisition is complete, the signal can be processed with options from the “Process” tab in the lower main window. In the “Default” submenu, the “Transform All” button can be used to Fourier Transform the FID signal and display the resulting spectrum in the upper main

window. In the “Weighting” submenu the line broadening (Hz) can be adjusted in the “Weight Parameters” list. After verifying proper data acquisition, use File>Save As...to save the details of the experiment setup as well as the FID data obtained.

Appendix B

Bruker X-ray diffractometer operation

B.1 Sample Preparation

Once a suitable set of chips is obtained by dicing a wafer of possibly questionable orientation a sample can be prepared for orientation confirmation by x-ray diffractometry as follows:

- (1) Score and cleanly break a glass rod so the length is about twice that of the chip
- (2) Using a soldering iron melt a very small amount of wax onto the back of the chip
- (3) Quickly attach the glass rod centered on the sample and parallel to the long edge (or parallel to the axis of rotation in the goniometer)
- (4) Melt enough wax into the sample holder so that the glass rod stays firmly in place when pushed gently into the softened (cooling) wax
- (5) Ensure that the axis of the glass rod is aligned with the axis of the sample holder so there is no wobbling during the rotation experiment

B.2 Starting the Bruker

- (1) Turn on cooler (to the right of machine)
- (2) Open water valve near sink
- (3) Turn on goniometer (switch at back of controller unit, 2nd box up from ground in machine cabinet)

Pause in Operation (days)	High Voltage/Duration							Total Duration
	20kV	25 kV	30 kV	35 kV	40 kV	45 kV	50 kV	
0.5 to 3	30 sec	30 sec	30 sec	30 sec	30 sec	30 sec	1 min	4 min
3 to 30	30 sec	30 sec	2 min	2 min	5 min	5 min	10 min	25 min
>30 or new x-ray tube	30 sec	30 sec	2 min	2 min	5 min	10 min	15 min	35 min

Table B.1: Voltage increment durations

- (4) Pull red knob on x-ray controller panel
- (5) Push and hold button between <off> and <on> until the green light comes on (a few seconds)
- (6) Push the <on> button
- (7) Use Crystal>Generator in SMART software window to increase the voltage and then the current according to Table B.1

B.3 Acquiring rotation data with SMART

Note: Operating temperature shown on front of goniometer controller should be about -54°C , which takes roughly 15 mins to obtain.

- (1) To align the sample in the goniometer use Goniom>Manual and then use the manual controls to rotate the sample around the ϕ -axis and adjust the sample with the Allen wrench as needed so the sample is roughly centered in the path of the beam during the whole rotation.
- (2) To run the experiment use Acquire>Rotation with the following settings:
 - Exposure time: 60 sec
 - Starting ϕ is unimportant (use 0 for consistency)
 - Use rotation axis 3 (ϕ -axis)

- (3) Save image with Acquire>Save Still...
- (4) To correct for the shape of the CCD detector (among other things) use Detector>Unwarp and save the unwarped image in a convenient location (.rot file extension)
- (5) To convert the rot file into a more useful tiff file format use the program fit2d (See B.5).

B.4 To turn off

- (1) Use Crystal > Generator again to lower the voltage and the current back down to 20 kV and 5 mA respectively.
- (2) Push the <off> button on the x-ray controller panel
- (3) Push the red knob
- (4) Turn off the goniometer
- (5) Close water valve
- (6) Turn off cooler

B.5 Using fit2d to create a tiff image

Note: The version of fit2d used in this project is V16.041

- (1) Open the program and click “I ACCEPT” to agree to the terms & conditions
- (2) Settings for “DIMENSIONS OF PROGRAM ARRAYS” page:
 - First dimension of arrays: 512
 - Second dimension of arrays: 512
 - Create memory arrays: YES
 - Create variance arrays: NO

- (3) Select “OK” and then “POWDER DIFFRACTION (2-D)” from the next menu
- (4) Select “Input” and then navigate to the unwarped rot file and select it then select “Bruker” file format
- (5) Settings for “DISTORTION CORRECTIONS” page:
 - Subtract dark current image: NO
 - Name of dark current file: (default)
 - Apply flat field correction: NO
 - Name of flat-field file: (default)
 - Apply scaling after flat-field correction: NO
 - Flat-field multiplier to apply: 1000
 - Apply spatial distortion correction: NO
 - Name of spatial distortion file: (default)
- (6) Select “OK” and now you should see an image
- (7) To clean up the image for ease of comparison to the CrystalMaker simulation (See Appendix C), select the “Z-SCALING” option from the menu and use the “+/- MAXIMUM/MINIMUM” buttons to adjust the image so there is a clear contrast between the spots and the image background. Try using the “WEAK PEAKS” option if you are having trouble. When the image spots and background are clearly contrasted, use the “USER MINIMUM/-MAXIMUM” buttons to take note of the minimum and maximum values of the image, as they will be useful in a moment. Click “EXIT”
- (8) Select “OUTPUT” at the main image menu to save the image, and then select the “TIFF 16 BIT” file format. When it asks about the file name select “NO” to choose the output location explicitly (otherwise it will end up somewhere unpredictable). Once an appropriate save location and file name have been selected press the [return] key. Enter the values

for the “USER MINIMUM/MAXIMUM” which were recorded earlier when asked for the “LOWER/UPPER LIMIT OF RANGE”. Select “OK” after each prompt.

- (9) Use a file browser to confirm that the file saved correctly before using “EXIT”>“EXIT FIT2D”>“YES” to confirm exit to fit2d.

Appendix C

The crystal rotation method

Note: The version of CrystalMaker used for this project is CrystalMaker 9.2.3 for Windows. The crystal rotation method is useful for confirmation of material and axis of rotation in a sample by use of x-ray diffractometry. The results of the experiment can be compared to a simulation of a Lowe Plate (Front-Scattering) experiment to confirm both composition and axis of rotation. Two files are needed before starting the method: a tiff image file which has been obtained by diffractometry as in Appendix B, and a compatible CrystalMaker data file (.cmdf or .cif extension) which contains the Bravais lattice structure of the sample under consideration. The axis of rotation used in the experiment outlined in Appendix B should be noted or guessed.

- (1) Use CrystalMaker to open the data file. Now use Measure>Powder Diffraction to obtain results for a simulation of the powder spectrum of the crystal sample, taking note of the orientation of the few smallest pertinent 2θ angles (hover mouse over peak to give orientation).
- (2) Use Measure>Single-Crystal Diffraction to open SingleCrystal. It is convenient to resize windows at this point so that the CrystalMaker and SingleCrystal windows are both visible. In SingleCrystal use Picture>Choose Picture... and navigate to the tiff file generated by the methods in Appendix B
- (3) In SingleCrystal make sure the Laue Plate (Front-Scattering) experiment type is used by selecting the option from the Diffract menu. Also under the Diffract menu, make sure that

the Wavelength range is (for Si at least) 0.68 to 0.74 Å. Select the Pic option to modify the picture, and then shift-click-and-drag around until the spots in the picture are aligned vertically or the part of the image that is missing is about horizontal. Use the Grid option to help with alignment.

- (4) In CrystalMaker orient the crystal so that the axis of rotation used in Appendix A is vertical in the plane of the screen. select Transform>Auto Rotate to rotate the crystal about the Y-axis in very small increments (0.1° - 0.2°).
- (5) When one of the spots generated by the simulation with a small index (recognized from the powder diffraction simulation) is seen use the Stop Rotation option in the toolbar to pause the rotation. In SingleCrystal click the Sim option in the toolbar and resize the simulation so the simulated spot falls on top of the spot in the tiff image.
- (6) Start the crystal rotation again in CrystalMaker and as each spot in the simulation appears check that it corresponds to a spot in the image file (with no exceptions!). Having a printed copy of the image file makes checking spot locations easier.

Once each spot in the simulation has been identified with each spot in the image file with no exceptions then the axis of rotation for the SingleCrystal simulation is confirmed to be the same as the one used in the experiment in Appendix B.

Appendix D

Radia code

The script presented here has been adapted from Mathematica notebook format to L^AT_EXformat.

Introduction

Model to infer susceptibility matching of a solid rectangular prism in a fluid.

Evaluate sections in the order of presentation: “Load and Initialize Radia” followed by “Build the Geometry”, “Plot the Geometry”, “Plot the Magnetic Field” ...

All sections must be evaluated in the order of presentation. A section may be evaluated several times with the same or different parameters if the previous sections have been evaluated before. The only limitation is memory. The Radia.exe memory can be re-initialized by re-executing the section entitled “Load and Initialize Radia”. The Kernel memory can be re-initialized by calling `Exit[]` and then re-executing all sections of this example. If the Front-End is running out of memory, close some windows or close the Front-End and start it again. Note that if by mistake any section is executed before the previous ones are executed, it may be necessary to exit the Kernel and re-start everything from the very beginning.

Load and Initialize Radia

The following instruction loads the Radia package and returns the Radia version number.

```
<< Radia`;
```

```
Radia Version: 4.3 is loaded
```

Radia is copyright ESRF, France.

Portions copyright Synchrotron SOLEIL, France.

Portions copyright Wolfram Research, Inc.

Control Parameters

Specification for geometry, susceptibility, field grid, spectrum bins and line broadening:

dChi is the solid volume susceptibility minus the liquid volume susceptibility in SI units. Both susceptibilities are diamagnetic (negative) so if dChi is positive the solid is less diamagnetic than the liquid, which is the usual situation before paramagnetic ions are added to the liquid.

$$\text{dChi} = 0.2 * 10^{-6};$$

fLsolid and fLliquid is the Larmor frequencies in Hz. They are needed to convert Bz field distortion to frequency units. The main NMR field is in the +z direction.

$$\text{fLsolid} = 99.362 * 10^6; \text{silicon-29}$$

$$\text{fLliquid} = 500.0 * 10^6; \text{proton}$$

The solid is a rectangular prism centered on the origin. Dimensions are in mm.

$$\text{solidX} = 2.5;$$

$$\text{solidY} = 2.5;$$

$$\text{solidZ} = 4.0;$$

The NMR tube is a cylinder with axis in the z direction and centered on the origin. Dimensions are in mm.

$$\text{tubeRadius} = 2.0;$$

$$\text{tubeLength} = 16.0;$$

number of field points in each direction

$$\text{nX} = 40; \text{nY} = 40; \text{nZ} = 160; (256,000 \text{ points})$$

$$\text{nPoints} = \text{nX} * \text{nY} * \text{nZ};$$

$$\text{Print}[\text{nPoints}, \text{"field points"}];$$

number of histogram bins for computing line shape

```
bins=600;
```

Lorentzian line broadening for liquid line shape, in Hz

```
liquidFWHM=5;
```

Frequency plot range for solid and liquid, in Hz

```
plotRangeSolid={0,40};
```

```
plotRangeLiquid={100,100};
```

256000 field points

Build the Geometry

The sample is a rectangular prism aligned with the z-direction, magnetized in the z-direction, and coaxial with the cylindrical NMR tube which is not magnetized. Dimensions in millimeters. Unit magnetization is used for the calculation, and then the results are scaled at the end.

```
radObjRecMag[{x,y,z},{wx,wy,wz},{mx,my,mz}:{0,0,0}]
```

```
radObjCylMag[{x,y,z}, r, h, nseg, a : "z", {mx,my,mz}:{0,0,0}]
```

Create the objects:

```
solid=radObjRecMag[{0,0,0},{solidX,solidY,solidZ},{0.0,0.0,1.0}]
```

```
tube=radObjCylMag[{0,0,0},tubeRadius,tubeLength,20,z,{0.0,0.0,0;0}]
```

```
Grp=radObjCnt[{solid,tube}]
```

These functions check if a point {x,y,z} is inside the solid or the tube

```
inTube[r_]:= (r[[1]]^2 + r[[2]]^2 < tubeRadius^2 && (Abs[r[[3]]] < tubeLength/2;
```

```
inSolid[r_]:= Abs[r[[1]]] < solidX/2 && Abs[r[[2]]] < solidY/2 && Abs[r[[3]]] < solidZ/2;
```

Plot the Geometry

These instructions plot the geometry.

Save the Geometry in “dr” and set opacity of all objects

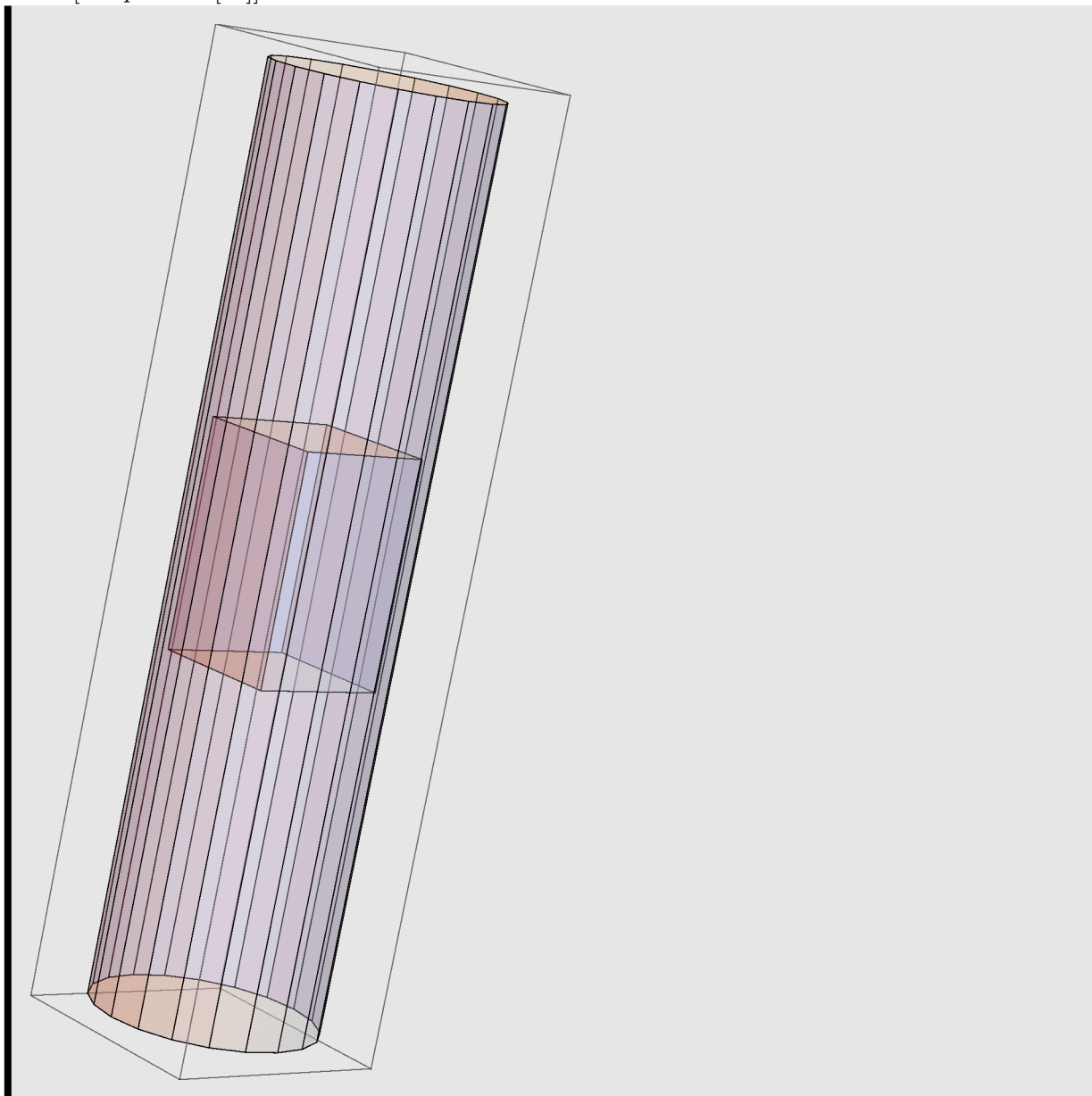
```
dr=radObjDrw[Grp];
```

```
dr={Opacity[0.3],dr}
```

Save the Geometry in “dr” and set opacity of all objects

Draw the geometry

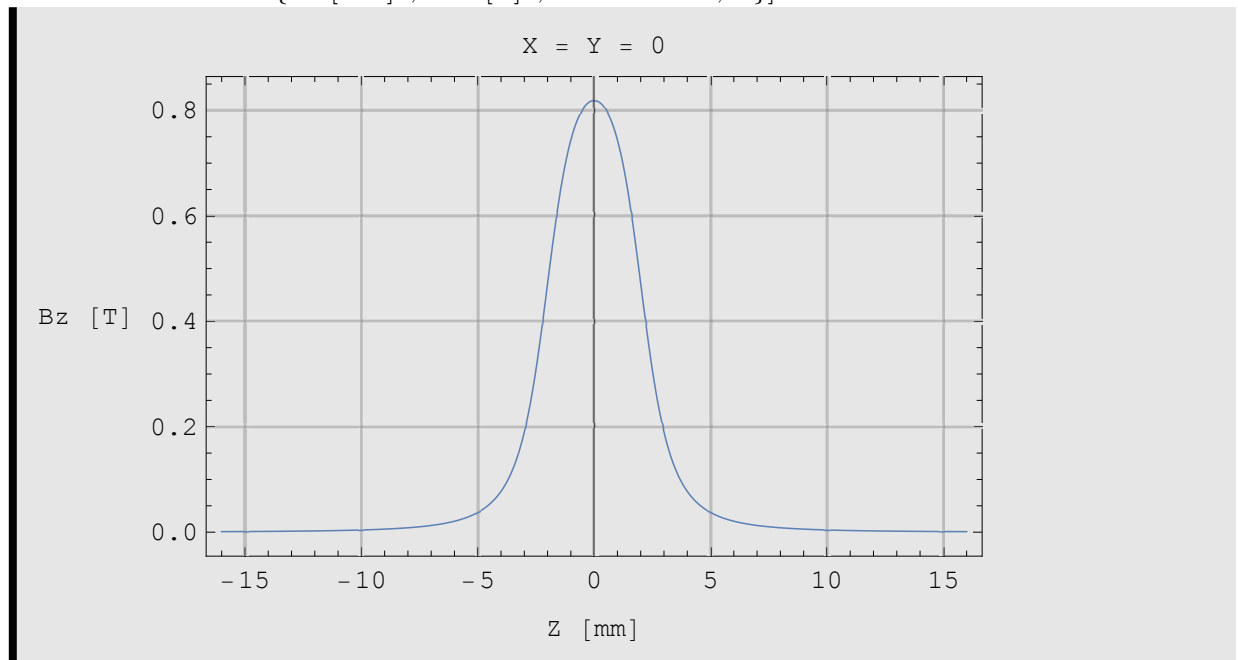
```
Show [Graphics3D[dr]]
```



Plot the Magnetic Field Along a Line

The magnetic field vertical component profile is plotted along the y axis. This is the field due to unit positive magnetization of the sample. `Plot[[Grp,"Bz",{0,0,z}],{z,-10,10},...]` fails. However, the `radFld2` function defined below works. This seems to be because `Plot` sends `Null` arguments.

```
radFld2[index_?NumericQ,fieldID_,{xc_?NumericQ,yc_?NumericQ,zc_?NumericQ}]:=
  radFld[index,fieldID, {xc,yc,zc}]
RadPlotOptions[];
Plot[radFld2[Grp,"Bz",{0,0,z}],{z,-tubeLegth,tubeLength},AxesOrigin-> {0,0},
  FrameLabel -> { "Z [mm]", "Bz [T]", "X = Y = 0", ""}]
```



Find the Lineshape Interior and Exterior to the Sample

A histogram of B_z over the volume gives the line shape.

```
nPoints=nX*nY*nZ;
Print[nPoints,"field points"];
```

Make lists of field values for each region over the grid

```
fieldValuesSolid={};
```



```

fieldValuesLiquid={};
For[i=1,i ≤ nX, i++, For[j=1, j ≤ nY, j++, For[k=1,k ≤ nZ, k++,
    r={2*tuberadius*(((i-1)/(nX-1))-1/2),2*tubeRadius*(((j-1)/(nY-1))-1/2),
    tubeLength*(((k-1)/(nZ-1))-1/2)}}]
If [inSolid[r], fieldValuesSolid=Append[fieldValuesSolid,radFld[Gpp,"Bz",r]];
If[inTube[r]&& !inSolid[r], fieldValuesLiquid=Append[fieldValuesLiquid,radFld[Grp,"Bz",r]];
Make histograms, bins run from -1 T to +1 T. Convert lists of bin boundaries to lists of bin centers
for plotting and convolution

```

```

histSolid=HistogramList[fieldValuesSolid, {-1.0,1.0,2.0/bins}];
histLiquid=HistogramList[fieldValuesLiquid,{-1.0,1.0,2.0/bins}];
binCenters=Range[Length[histSolid[[1]]]-1];
For[i=1,i< Length[histSolid[[1]]],i++,binCenters[[i]]=histSolid[[1,i]]+histSolid[[1,i+1]])/2]
histSolid[[1]]=binCenters;
histLiquid[[1]]=binCenters;

```

At his point, bin centers are in Tesla for unit magnetization. Convert to Hz.

```

histSolid[[1]]=fLsolid*dChi*histSolid[[1]];
histLiquid[[1]]=fLliquid*dChi*histLiquid[[1]];

```

Construct Lorentzian kernal for broadening line in the liquid. Make it symmetric with an odd number of points so that the maximum is at the center. Use enough points so that it decays almost to zero and normalize it.

```

binStep=2*fLliquid*dChi/bins;

```

bin width in Hz

```

kernPoints=2*Floor[3*liquidFWHM/binStep]+1;

```

enough points and odd

```

lorntzianKern=Table[1.0/((liquidFWHM/2.0)^2+(i*binStep)^2,
    {i,-(kernPoints-1)/2,+(kernPoints-1)/2}];
lorentzianKern=lorntzianKern/Total[lorntzianKern];

```

normalize it

```
lorentzianFreq=Table[i*binStep,{i,-(kernPoints-1)/2,+(kernPoints-1)/2}];
```

frequency list to plot the kernal

Plot the line broadening kernal

```
ListPlot[Transpose[{lorentzianFreq,lorentzianKern}],Frame -> True, Axes -> False,
PlotRange -> All, FrameLabel -> {"Frequency (Hz)", "", "Lorentzian Kernal" }]
```

Construct the broadened liquid line shape

```
histLiquidBroadened=histLiquid;
histLiquidBroadened[[2]]=ListCorrelate[lorentzianKern,histLiquid[[2]],(kernPoints+1)/2];
```

Find the linewidth characterized by the RMS deviation

```
fBar=Mean[fieldValuesSolid]*fLsolid*dChi;
```

```
lineWidthSolid=
```

$$\sqrt{\left[\frac{1}{\text{Length}[\text{histSolid}[[1]]]} \sum [\text{histSolid}[[2, m]] * (\text{histSolid}[[1, m]] - \text{fBar})^2, \{m, 1, \text{Length}[\text{histSolid}[[1]]\}] \right]};$$

Use median frequency to determind RMS deviation in histogram and determine ‘characteristic’

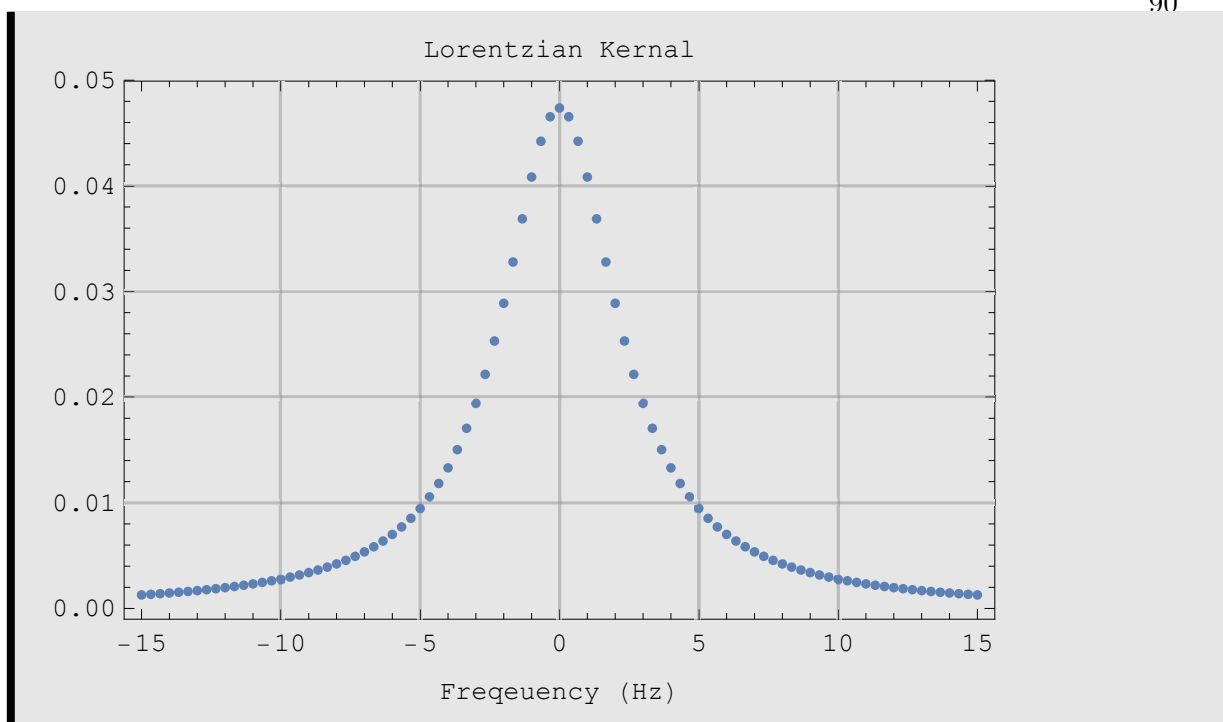
linewidth

```
Pring[lineWidthSolid,“Hz RMS Frequency Linewidth Characterization”];
```

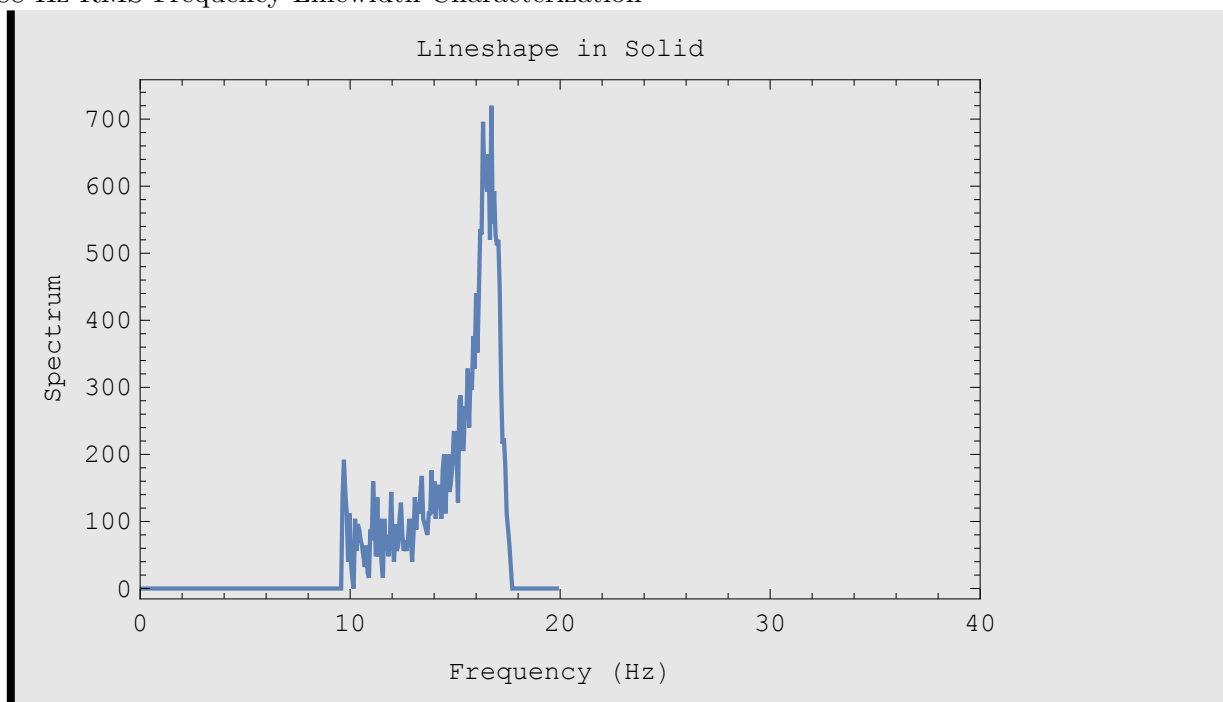
Plot the solid and liquid line shapes

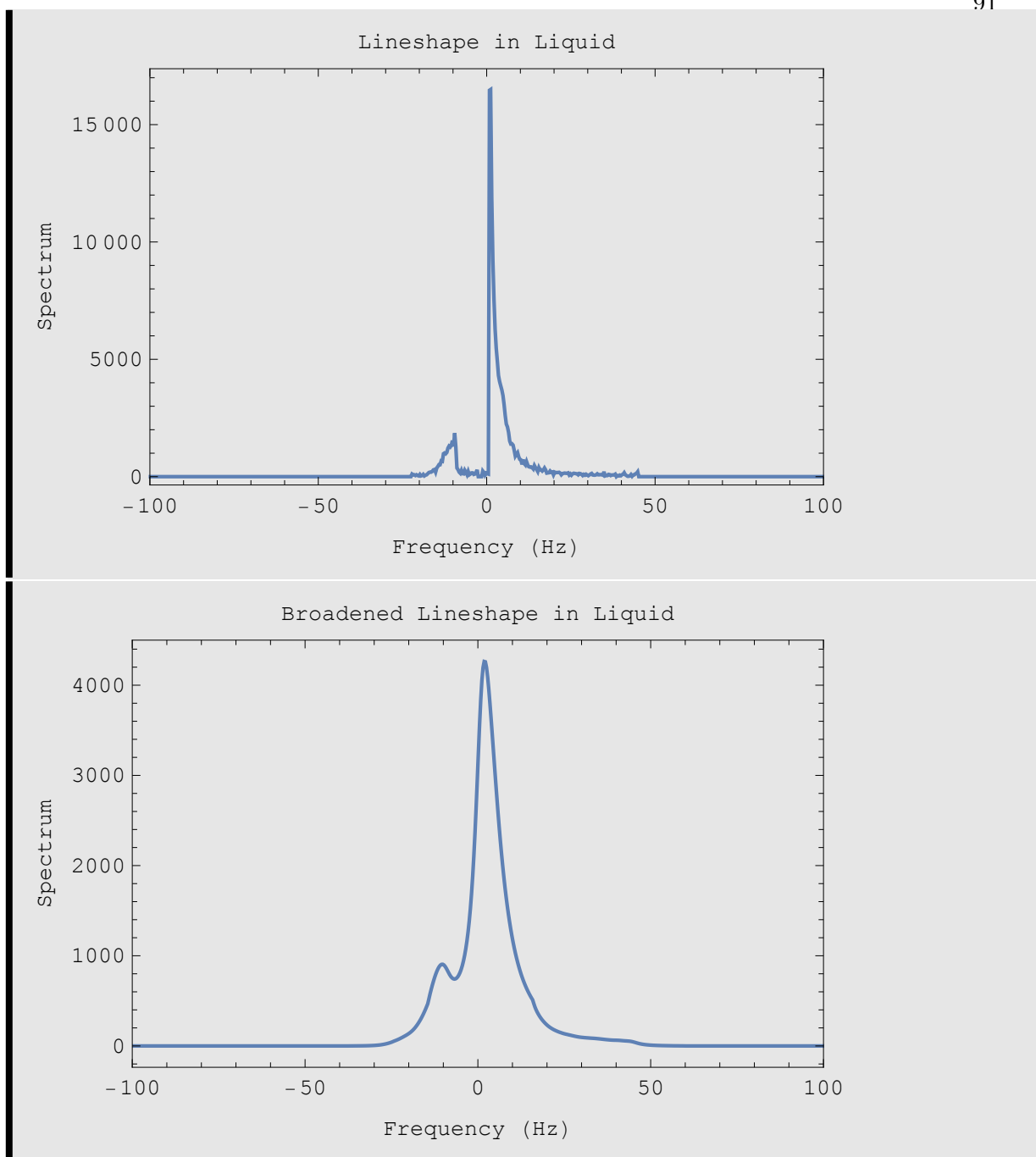
```
ListLinePlot[Transpose[histSolid],Frame -> True, Axes -> False,
PlotRange ->plotRangeSolid, All},FrameLabel -> “Lineshape in Solid”}]
ListLinePlot[Transpose[histLiquid],Frame -> True, Axes -> False, PlotRange -> {plotRangeLiquid,All},
FrameLabel -> “Frequency (Hz)”,“Spectrum”,“Lineshape in Liquid”}]
ListLinePlot[Transpose[histLiquidBroadened],Frame -> True, Axes -> False,
PlotRange -> {plotRangeLiquid,All},FrameLabel -> “Frequency (Hz)”,
“Spectrum”,“Broadened Lineshape in Liquid”}]
```

256000 field points



12.7288 Hz RMS Frequency Linewidth Characterization





Plot Field Grid

```
nPoints=nX*nY*nZ;
```

```
Print[nPoints,"field points"];
```

```
pointsSolid={};
```

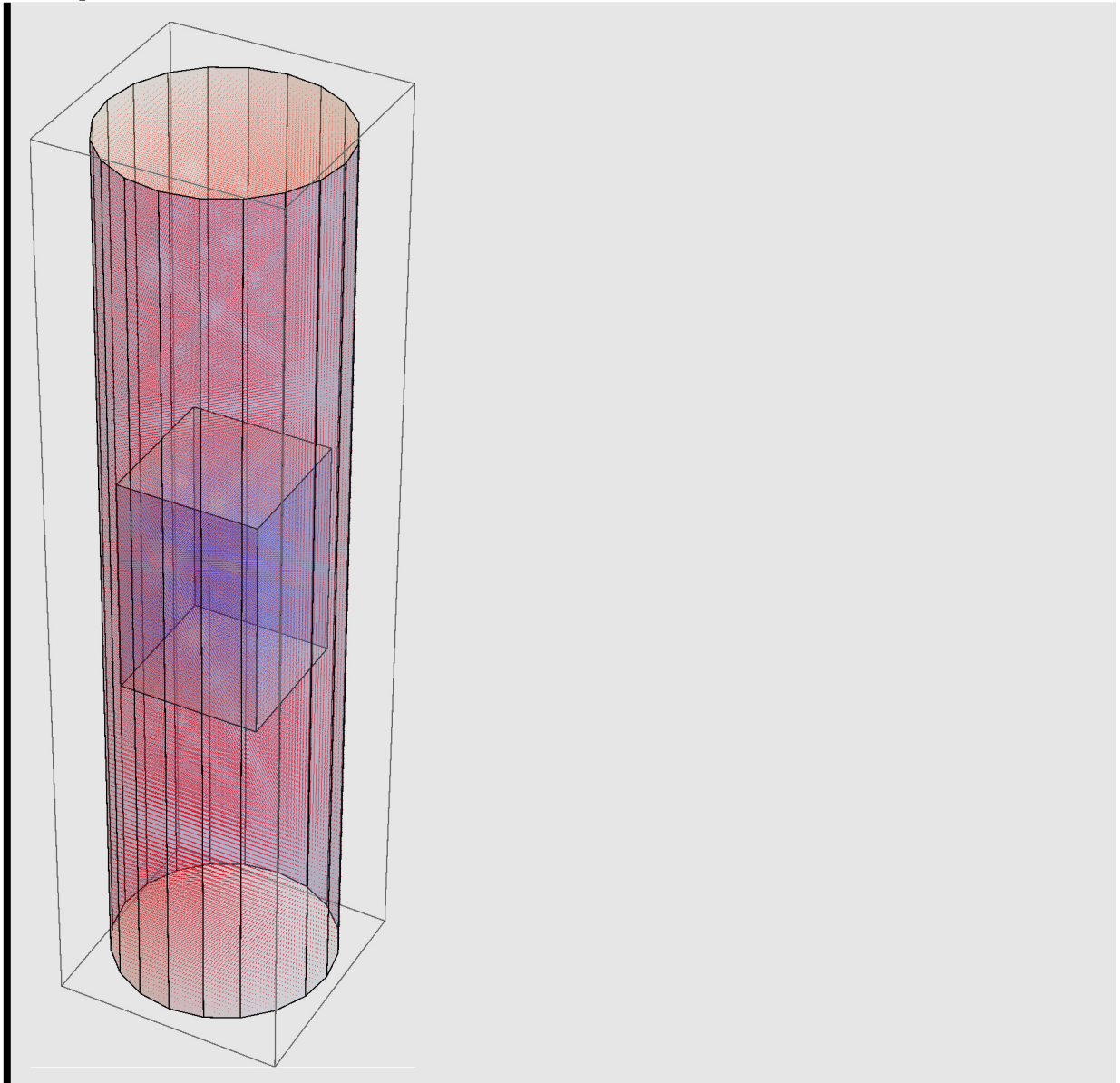
```
pointsLiquid={};
```

```

For[i=1,i ≤ nX, i++, For[j=1,j ≤ nY, j++, For[k=1,k ≤ nZ, k++,
  r={2*tubeRadius*(((i-1)/(nX-1))-1/2),2*tubeRadius*(((j-1)/(nY-1))-1/2),
  tubeLength*(((k-1)/(nZ-1))-1/2)}]]];
If[inSolid[r],pointsSolid=Append[pointsSolid,r]];
If[inTube[r] && !inSolid[r],pointsLiquid=Append[pointsLiquid,r]];
Graphics3D[{dr, Blue, PointSize[0.003],Point[pointsSolid],Red,Point[pointsLiquid]}]

```

256000 field points



Appendix E

Spinach code

E.1 Silicon simulation main body code

```
% silicon/diamond lattice dilutely occupied with spin -1/2
% generate disorder configurations and fids
%
% Spinach IK-0 basis set, all products of up to 'level' spins
% set 'level' >= number of spins for exact Liouville simulation
%
% John Price and Brooks Christensen
% Date: 071816
%
% Simulate fid of randomly populated dilute 3d lattice of spins
% Single spin type with secular dipole interactions
% Axisymmetric nearest-neighbor J-coupling tensor
% Silicon Bravais lattice vectors and basis
% Periodic boundary conditions
%
% Be sure to set keep='all' to simulate all spins in the configuration
%
```

```

% john.price@colorado.edu

% Control parameters

par.serialStart=4; % file serial number start
par.ranSeedStart=par.serialStart; % starting seed for random generator
par.throws=1; % number of disorder configurations
par.level=10; % basis order
par.minCellDimensionVector=[4,4,4];
par.maxCellDimensionVector=[4,4,4];
par.cutoff_distance=15.0;
% cutoff distance in A for dipole interactions
% Si nearest neighbor dist. is 2.3517
par.cutoffJ=3.0; % J tensor cutoff in A
% nearest neighbor distance is 2.352 A and next NN is 3.840 A
jIso=70; %Hz
deltaJ=90; %Hz
par.axialJ=(3*jIso+2*deltaJ)/3; % axial J coupling in Hz
par.transJ=(3*jIso-deltaJ)/3; % transverse J coupling in Hz
par.Bfield=11.743; % 11.743 Teslas -> 500 MHz for protons
par.isotope='29Si'; % isotope
par.a0=5.43096; % Si crystal lattice constant
par.filling_fraction=0.046832; % filling fraction
par.fwhm=10.0; % exp line broadening in Hz for plots

% keep is a vector of spin labels to include (positive integers)
% if keep='all' include all generated spins
% run the code first with keep='all', throws=1

```

```

% and a certain value of ranSeedStart

% examine the geometry plot and decide which spins to keep

% set keep and run again

keep='all ';

if strcmp(keep,'all ')                                % fixes problem with reassigning
    keepFlag=1;                                       % keep inside of loop
else
    keepFlag=0;
end

% define lattice for standard ([100]) orientation – rotated after H is
%   calculated

% Bravais lattice vectors
a1=par.a0/2*[1,1,0];
a2=par.a0/2*[1,0,1];
a3=par.a0/2*[0,1,1];
% Lattice basis
bravais_basis={ [0,0,0], (par.a0/4)*[1,1,1] };
bravais={a1,a2,a3};

par.cellVolume=dot(a1,cross(a2,a3));

% build the diagonal J-coupling tensor (principal axis eigenbasis)
principalValueDiagMat=[...

```



```

    par.transJ 0 0;...           % 3rd eigenvalue should be axial if
    0 par.transJ 0;...           % rotation is the identity because
    0 0 par.axialJ];             % then axial is cartesian z

% loop over disorder configurations
runTimes=zeros(1,par.throws);    % variable to save run times
numSpinsArray=zeros(1,par.throws); % variable to save number of spins
cellDimArray=cell(1,par.throws);
% variable to save cell dimensions for normalization
serial=par.serialStart;          % initialize file serial number
multVecFlag=0;
for stepNum=1:par.throws
    tic;                          % time each iteration
    ranSeed=stepNum-1+par.ranSeedStart; % unique seed for each iteration

    if keepFlag==1
        keep='all';
    end

    % Create cell array of spin coordinates, also coords of all sites
    [coords,nums,par,sites,keep]=...
        lattice_builder_v1(par,bravais,bravais_basis,ranSeed,keep);
    num_spins=nums.num_spins;

    % Create J-coupling interaction matrix for Spinach
    [jCouplingMatrix,pMatCellArray,multVecMatrix]=...
        j_coupling_generator_v2(...

```

```

num_spins , par , bravais , coords , principalValueDiagMat );

if num_spins>0

% % Build data structures for Spinach
sys.output=['logFile ', num2str( serial ), '.txt '];
sys.magnet=par.Bfield;
sys.tols.prox_cutoff=par.cutoff_distance;
inter.coordinates=coords;           % spin coordinates
sys.isotopes=cell(1,num_spins);      % isotope list
sys.isotopes(:)={par.isotope};
scalar=cell(1,num_spins);           % interaction Zeeman terms
scalar(:)={0};
inter.zeeman.scalar=scalar;          % zemann term in ppm of B field
inter.coupling.matrix=jCouplingMatrix; % jCouplingMatrix
inter.temperature=298;

% % Periodic boundary conditions
inter.pbc={par.cells_a*bravais{1}...
           par.cells_b*bravais{2} par.cells_c*bravais{3}};

% % Basis set
bas.formalism='sphten-liouv';
bas.approximation='IK-0';           % restrict basis
bas.connectivity='full_tensors';
bas.projections=[-1 1]; % only these coherence orders are present
spin_system=create(sys,inter);

```

```

%% % Report

report(spin_system,['User: disorder config. no. ' num2str(stepNum)])
report(spin_system,['User: ' num2str(nums.num_cells) ...
    ' Bravais lattice cells '])
report(spin_system,['User: ' num2str(nums.num_sites) ' spin sites '])
report(spin_system,['User: ' num2str(par.filling_fraction*...
    nums.num_sites) ' mean occupied sites '])
report(spin_system,['User: ' num2str(num_spins) ' occupied sites '])

%% % Restrict basis

level=par.level;

if num_spins<par.level
    level=num_spins;
end

bas.level=level;

% Spinach housekeeping

spin_system=basis(spin_system,bas);

% Set initial state to Mx

rho=(state(spin_system,'L+',par.isotope)...
    +state(spin_system,'L-',par.isotope))/2;

% Sequence parameters

parameters.spins={par.isotope};

parameters.rho0=rho;

parameters.coil=state(spin_system,'L+',par.isotope);

```

```

parameters.decouple={};
parameters.offset=0;
parameters.sweep=5000;
parameters.npoints=500;
parameters.zerofill=20000;
parameters.axis_units='Hz';
parameters.invert_axis=1;
parameters.grid='si_4_orientations';

% Simulation
fids=powder(spin_system,@acquire,parameters,'nmr');

for n=1:size(fids,1)
    % Save unapodized fid
    fid=fids{n};
    save(['Si_fid ' num2str(serial) '_' num2str(n)], 'fid')
end

else
    fid=zeros(parameters.npoints,length(fids));
    save(['Si_fid ' num2str(serial) '_' num2str(n)], 'fid')
end

serial=serial+1;
numSpinsArray(stepNum)=num_spins;
cellDimArray{stepNum}=[par.cells_a,par.cells_b,par.cells_c];
runTimes(stepNum)=toc;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cluster_plotter_v1(coords,sites,num_spins,nums,keep,...
    principalValueDiagMat,pMatCellArray,multVecMatrix)
end

runTimeSum=sum(runTimes);
save('Si_master_log','par','runTimeSum')
save(['Si_log' num2str(par.serialStart) '-' num2str(serial-1)],...
    'runTimes','numSpinsArray','par','coords',...
    'pMatCellArray','jCouplingMatrix','cellDimArray')
fidSum_v5

```

E.2 Lattice builder function code

```

% Brooks Christensen and John C. Price
% v1: September 8th, 2016
% function name: lattice_builder_vXX.m
% function takes arguments par, bravais, bravais_basis, ranSeed, and keep, and
% returns the coordinates in a 1x(num_spins) cell array of 1x3 coordinate
% vectors and the number of spins, sites, and cells given as a structure
% array with fields num_spins, num_sites, and num_cells. Also returns
% par to update number of cells per bravais lattice axis

function [coords,nums,par,sites,keep,zeemanMatrix]=lattice_builder_v2(...
    species,par,bravais,bravais_basis,ranSeed,keep)

```

```

nums.num_spins=0;

nums.num_sites=0;

nums.num_cells=0;

if strcmp(species,'SiO2')

    diagonalMatrix=...

        [-102.6  0  0;...

         0 -107  0;...

         0  0 -109.1];

    eigenvectorMatrix=...

        [0.008  0.997  0.029;...

         0.559  0.027  0.828;...

         0.828  0.010 -0.557];

    rotationMatrix1=...

        [-1/2 -sqrt(3)/2  0;...

         sqrt(3)/2 -1/2  0;...

         0  0  1];

    rotationMatrix2=...

        [-1/2  sqrt(3)/2  0;...

         -sqrt(3)/2 -1/2  0;...

         0  0  1];

    rotMat=[1  0  0;0  1  0;0  0  1];

    zeemanMatrices=...

        {(rotMat*eigenvectorMatrix)*diagonalMatrix*...

         (rotMat*eigenvectorMatrix)^-1,...

         (rotMat*(rotationMatrix1*eigenvectorMatrix))*diagonalMatrix*...

```

```

        (rotMat*(rotationMatrix1*eigenvectorMatrix))^ -1,...
        (rotMat*(rotationMatrix2*eigenvectorMatrix))*diagonalMatrix*...
        (rotMat*(rotationMatrix2*eigenvectorMatrix))^ -1};

end

cellSizeVec=par.minCellDimensionVector;          % increases dimensions
% one by one until either max spins or
% max dimension size is reached first
num_sites_temp=length(bravais_basis)*...
        cellSizeVec(1)*cellSizeVec(2)*cellSizeVec(3);
% finds first num_sites_temp value
rng(ranSeed);          % seed the random generator
filled=rand(1,num_sites_temp);          % finds first filled vector
filled=filled<par.filling_fraction;      % 1 with prob = filling_fraction
dimSelector=3;          % selects axis for incrementing cellSizeVec
inputFlag=0;

if sum(filled)>par.level
    inputFlag=1;
end

while ((sum(filled)<=par.level)||inputFlag)&&...
    (cellSizeVec(1)<=par.maxCellDimensionVector(1))&&...
    (cellSizeVec(2)<=par.maxCellDimensionVector(2))&&...
    (cellSizeVec(3)<=par.maxCellDimensionVector(3))

    % Create cell array of spin coordinates, also coords of all sites

```

```

coords={}; % spin coords
sites={}; % coords of all sites
zeemanMatrix={}; % tensor matrices at each filled site
num_spins=0; % count number of spins
site_index=0; % site index
for ini=0:cellSizeVec(1)-1
    for inj=0:cellSizeVec(2)-1
        for ink=0:cellSizeVec(3)-1
            for inm=1:length(bravais_basis)
                new_coord=ini*bravais{1}+inj*bravais{2}+...
                    ink*bravais{3}+bravais_basis{inm};
                sites=[sites new_coord]; % every site
                site_index=site_index+1 % bump site index
                if filled(site_index) % fill the sites randomly
                    coords=[coords new_coord]; % concat to cell array
                    num_spins=num_spins+1; % bump number of spins
                    if strcmp(species,'SiO2')
                        if inm==1 % create zeeman matrices
                            zeemanMatrix=...
                                [zeemanMatrix zeemanMatrices{1}];
                        elseif inm==2
                            zeemanMatrix=...
                                [zeemanMatrix zeemanMatrices{2}];
                        elseif inm==3
                            zeemanMatrix=...
                                [zeemanMatrix zeemanMatrices{3}];
                        end
                    end
                end
            end
        end
    end
end

```



```

                                end
                        end
                end
        end
end

end

num_sites=site_index;

cellSizeVec(dimSelector)=cellSizeVec(dimSelector)+1;
ds=dimSelector;
dimSelector=mod(dimSelector-1,3);
if dimSelector==0
        dimSelector=3;
end

% Create row vector identifying filled sites , 0=empty , 1=filled
num_sites_temp=length(bravais_basis)*...
        cellSizeVec(1)*cellSizeVec(2)*cellSizeVec(3);
rng(ranSeed);      % seed the random generator
filled=rand(1,num_sites_temp); % uniform from 0 to 1
filled=filled<par.filling_fraction; % 1 with prob = filling_fraction
end

cellSizeVec(ds)=cellSizeVec(ds)-1;
par.cells_a=cellSizeVec(1);
par.cells_b=cellSizeVec(2);
par.cells_c=cellSizeVec(3);

```

```

% num cells and num sites

nums.num_cells=par.cells_a*par.cells_b*par.cells_c;    % total num of cells

% keep only desired spins
if strcmp(keep,'all')

    keep=linspace(1,num_spins,num_spins);    % keep all spins
end

coords=coords(keep);                                % update coords to keep
nums.num_spins=length(keep);                        % update number of spins kept

```

E.3 J-coupling interaction function code

```

% Brooks Christensen and John C. Price
% v1: 9/8/16
% function name: j-coupling-generator_vXX.m
% This function takes as arguments num_spins, par, bravais, coords, and
% principalValueDiagmat. The function output is a (num_spins)x(num_spins)
% cell array of 3x3 J-coupling tensors called jCouplingMatrix

function [jCouplingMatrix,pMatCellArray,multVecMatrix]=...
    j-coupling-generator_v2(...
        num_spins,par,bravais,coords,principalValueDiagMat)

% Initialize (build) J-Coupling matrix for Spinach
%      n x n cell array of 3 x 3 matrices (initialized to zero)
jCouplingMatrix=cell(num_spins);
pMatCellArray=cell(num_spins);
for count1=1:num_spins

```

```

for count2=1:num_spins

    jCouplingMatrix{count1 ,count2}=zeros (3);

    pMatCellArray{count1 ,count2}=zeros (3);

end

end

% Build connectivity matrix for J-Coupling

conmatrix=false (num_spins );

multVecMatrix=cell (num_spins );

for index1=1:num_spins

    for index2=1:num_spins

        for multiplier1=-1:1

            for multiplier2=-1:1

                for multiplier3=-1:1

                    % create the multVec which points to the image of

                    % the vector translated into each of the 26

                    % immediately surrounding cells using multipliers

                    multVec=bravais{1}*par.cells_a*multiplier1+...

                        bravais{2}*par.cells_b*multiplier2+...

                        bravais{3}*par.cells_c*multiplier3;

                    % look for possible different configurations where

                    % assignments across boundaries have more than one

                    % possible arrangement (small configurations?)

                    if (~conmatrix(index1 ,index2))&&(index1~=index2)

                        % set conmatrix element to true if nearest

                        % neighbors across boundary

                        conmatrix(index1 ,index2)=...

```

```

        (norm(coords{index1}-coords{index2}+multVec)...
        <par.cutoffJ);

% use symmetry to assign the equal and
% opposite vector for nearest neighbors across
% the boundary between nearest cells
if conmatrix(index1,index2)
    multVecMatrix{index1,index2}=multVec;
    multVecMatrix{index2,index1}=-multVec;
end
elseif (norm(coords{index1}-coords{index2}+multVec)...
        <par.cutoffJ)&&...
        ~isequal(multVec,multVecMatrix{index1,index2})...
        &&(index1~=index2)
    multVecFlag='There is more than one multVec';
    disp(multVecFlag)
end
end
end
end
end

for counter1=1:num_spins
    for counter2=1:num_spins
        if conmatrix(counter1,counter2)
            if counter1>counter2
                % find normalized vector pointing from spin 1 to spin 2

```

```

rLength=norm( coords{ counter1 }-coords{ counter2 }+...
    multVecMatrix{ counter1 , counter2 } );
rVec=(coords{ counter1 }-coords{ counter2 }+...
    multVecMatrix{ counter1 , counter2 })/rLength;

% rotate principalValueDiagMat so that the transverse
% (unique) axis is aligned with the transverse axis of
% the bond between the two spins (only 2 rotations
% required)

% rotate about y-axis (RH convention assumed) by angle
% beta and then about z-axis by angle gamma
if rVec(3)>0
    beta=atan( sqrt( rVec(1)^2+rVec(2)^2 )/rVec(3) );
elseif rVec(3)<0
    beta=pi+atan( sqrt( rVec(1)^2+rVec(2)^2 )/rVec(3) );
else
    beta=pi/2;
end

if rVec(1)>0
    alpha=atan( rVec(2)/rVec(1) );
elseif rVec(1)<0
    alpha=atan( rVec(2)/rVec(1) )+pi;
elseif rVec(2)>0
    alpha=pi/2;
else

```

```

        alpha=-pi/2;
    end

    rotMat=euler2dcm(alpha,beta,0);

    jCouplingMatrix{counter1,counter2}=rotMat*...
        principalValueDiagMat*rotMat';

    [pMatCellArray{counter1,counter2},diagMat]=...
        eig(jCouplingMatrix{counter1,counter2});
    end
end
end
end
end
end

```

E.4 FID processing and spectrum plotting code

```

% Sum fids generated by siliconIKzero.m or siliconIKnone.m
%
% Raw fids are not normalized. Log file must be present so the number
% spins in each fid can be read in and used to normalized individual fids
% before the are summed.

load('Si_master_log')

% Control parameters
serialStart=par.serialStart; % starting input file serial number

```

```

throws=par.throws;           % number of files (number of disorder configs)

fwhm=10.0;                   % spectrum exp apodization, FWHM in Hz

% Get number of figures to produce based on number of orientations in
% powder grid file

gridFile='si_4-orientations'; % which grid file

rootDir=which('existentials'); % find root directory

rootDir=rootDir(1:end-32);    % truncate char array to root dir

spherical_grid=load([rootDir '/kernel/grids/' gridFile], 'weights');

% get weights

numFigs=length(spherical_grid.weights);

% find number of weights (which is same as number of figs)

% Sequence parameters (used when generating fids)

offset=0;                    % offset in Hz

sweep=5000;                  % sweep width = sampling rate in Hz

npoints=500;                 % number of fid points

zerofill=20000;              % zerofill points before fft

% Load log file and get number of spins for each fid (note: num_spins same
% for all orientations)

logData = load(['Si_log' num2str(serialStart) '-' ...
               num2str(serialStart+throws-1)]);

numSpinsArray=logData.numSpinsArray;

cellDimArray=logData.cellDimArray;

cellVolume=logData.par.cellVolume;

```

```

for counter=1:numFigs

% Sum the fids , normalize first point to number of spins
fidSummed=zeros(npoints,numFigs);
for index = 1:throws
    fid = load(['Si_fid' num2str(serialStart+index-1) '_' ...
               num2str(counter)]);
    cellNum=cellDimArray{index}(1)*cellDimArray{index}(2)*...
            cellDimArray{index}(3);
    clusterVolume=cellNum*cellVolume;
    if fid.fid(1) > 0
        normFid=fid.fid*numSpinsArray(index)/...           % first point =
            (fid.fid(1)*clusterVolume);
        % number of spins per unit volume
        fidSummed(:,counter)=fidSummed(:,counter)+normFid;
    end
end

% Apodize and Fourier transform
exp_rate=pi*fwhm*npoints/sweep;
fidSummed_filtered=cell(npoints,numFigs);
spectrum=cell(zeroFill,numFigs);

fidSummed_filtered{counter}=...
    apodization(fidSummed(:,counter),'exp-1d',exp_rate);
spectrum{counter}=fftshift(fft(fidSummed_filtered{counter},zeroFill));

```



```

% Normalize spectrum to total number of spins (for no particular reason)
spectrum{counter}=spectrum{counter}...
    *sum(numSpinsArray)/sum(real(spectrum{counter}));

% Plot spectrum and save the figure
figure
ax=linspace(-sweep/2,sweep/2,zeroFill)+offset;
ax_label='29Si offset linear frequency / Hz';
plot(ax,real(spectrum{counter}));
axis tight;
xlabel(ax_label);
set(gca,'XDir','reverse');
xlim([-1000 1000])
savefig(['Si_spec ' num2str(serialStart) '-' ...
    num2str(serialStart+throws-1) '-' num2str(counter) '.fig'])
end

% Save unapodized fidSum
save(['Si_fid ' num2str(serialStart) '-' ...
    num2str(serialStart+throws-1)],'fidSummed')

% Save spectrum
save(['Si_spec ' num2str(serialStart) '-' ...
    num2str(serialStart+throws-1)],'spectrum')

```